Carnegie Mellon University
The Robotics Institute

Carnegie Mellon University
School of Computer Science

Working Papers **Journal**

Volume 10 | Fall 2022

RISS | ROBOTICS INSTITUTE SUMMER SCHOLARS

Carnegie Mellon Robotics Institute Summer Scholars

Working Papers **Journal**

RISS Director
Dr. John M. Dolan
jdolan@andrew.cmu.edu

RISS Co-director
Ms. Rachel Burcin
rachel@cmu.edu

We gratefully acknowledge the support of the National Science Foundation through the Research Experience for Undergraduates (REU) program (Grant #1950811).

The Robotics Institute Summer Scholars Working Papers Journal is an annual publication of the Robotics Institute's Summer Scholars Program at Carnegie Mellon University.

Thank you to the incredible community who generously gave their time & expertise to guide and mentor the 2022 RI Summer Scholars.

The scholars would like to especially thank all those who helped to shape the work contained in this journal. We thank those listed below and the many more that have invested in the futures of these students.

Meet the 2022 RISS Community

Mentors

Aaron M. Johnson
Abhinav Gupta
Abhisesh Silwal
Akhil Padmanabha
Ardalan Tajbakhsh
Artur Dubrawski
Brady Moon
Chen Wang
Cherie Ho
Chirag Nagpal
Christoph Mertz
Dana Hughes
Deva Ramanan
George Kantor
Heng Yu
Ifigeneia Apostolopoulou
Isaac K. Isukapati
James McCann
Jean Oh
Jieshi Chen
John M. Dolan
Jonathon Luiten
Joseph Campbell
Junyi Geng
Katia Sycara
Kavya Puthuveetil
Khiem Vuong
Kyle Miller
Laszlo Jeni
Melisa Orta Martinez
Merritt Kowaleski
Micah Corah
Moonyoung Lee
N. Dinesh Reddy
Nicholas Gisolfi
Nikhil Varma Keetha
Oliver Kroemer
Pedro Morgado
Peter Schaldenbrand
Qin Lin
Raunaq Bhirangi
Robert Edman
Samantha Speer
Samuel Triest
Sarah Bergbreiter
Sebastian Scherer
Seth Karten
Shibo Zhao
Simon Stepputtis
Srinivasa Narasimhan
Stephen F. Smith
Sukjun Kim
Tess Hellebrekers
Tom Bu
Victoria Dean
Wenshan Wang
Xinyu Li
Yaoyu Hu
Yiwei Lyu
Yue Guo
Yufei Wang
Zachary B. Rubinstein
Zackory Erickson
Zeynep Temel

Workshop Partners & Presenters

Ander Solozano
Brad Eiben
Brady Moon
Carlotta Berry
Daniel Young
Fei Fang
Georgia Chalvatzaki
Howie Choset
Indu Panigrahi
Jessica Benner
Jordi Albo-Canals
Keely Austin
Liam Paull
Maja Matarić
Matthew Johnson-Roberson
Melisa Orta Martinez
Michael Milford
Neehar Peri
Rayna Hata
Sarah Young
Simon Lucey
Susan Finger
Tapomayukh Bhattacharjee
Teodor Tchalakov
Trent Tabor

Partner Offices

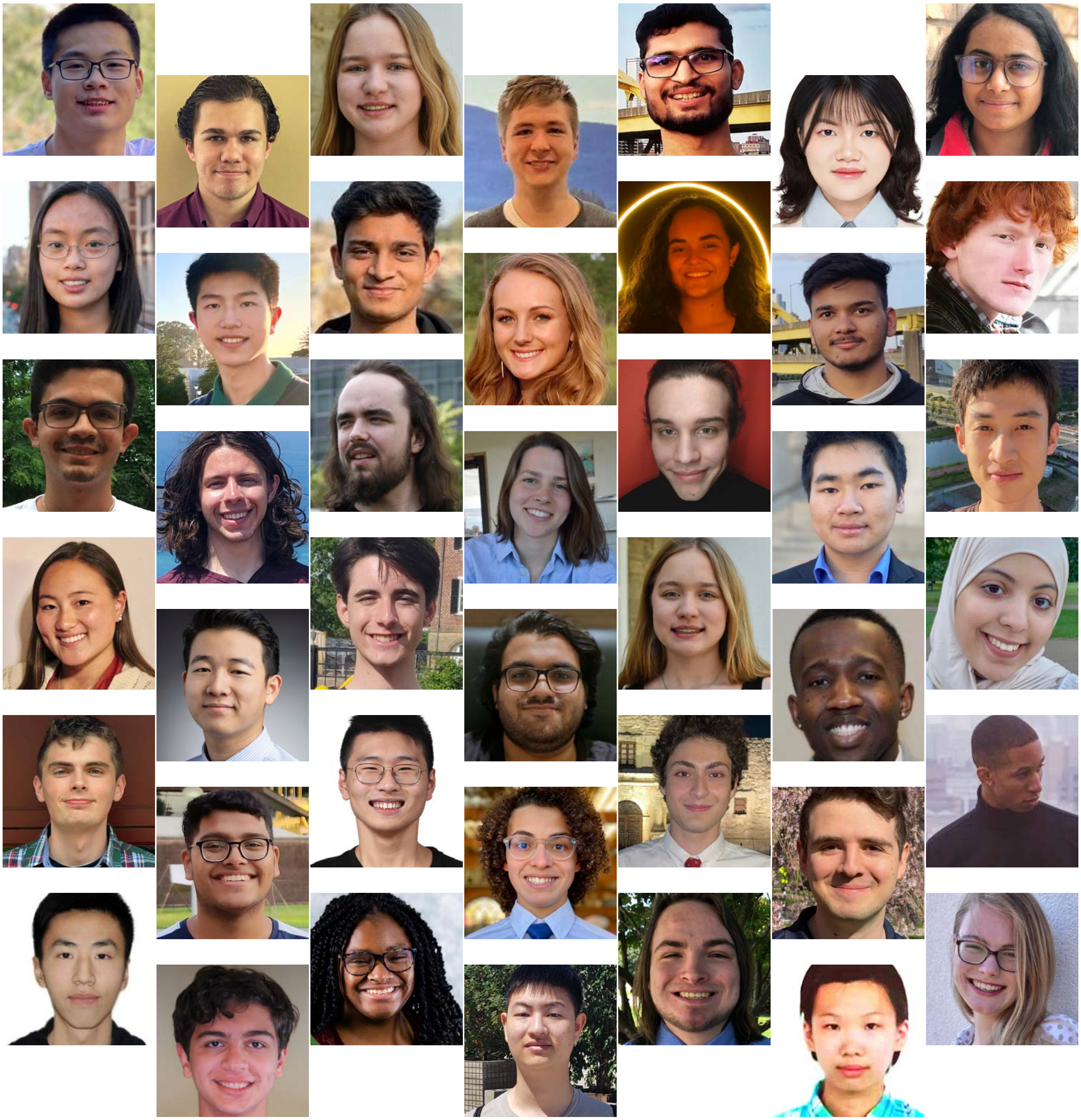
Robotics Institute, Office of the Director
School of Computer Science, Office of the Dean
TC Energy Foundation
Center for Student Academic Success
Traffic21
University Library Systems
CMU Business & Administrative Offices
Robotics Institute Business, Administrative, & Creative Media Offices
Center for Student Diversity & Inclusion
RoboOrg

Table of Contents

Table of Contents

Continued

47 SCHOLARS
FROM **11** COUNTRIES
& **40** UNIVERSITIES



Dear Colleagues

Welcome to the 10th Volume of the Carnegie Mellon Robotics Institute Summer Scholars (RISS) Working Papers Journal. We are pleased to present this collection of articles that reflects the range of topics and diversity of research to which the scholars contributed in the summer of 2022.

Founded in 2006, the CMU RISS program and community create opportunities for undergraduate students from across the country and around the world to explore robotics. A diverse and inclusive working and learning environment where all students are actively welcomed, included, and supported is fostered. Scholars build knowledge, skills, and a network that will open doors for years to come. RISS is part of a powerful coalition of stakeholders growing and diversifying the robotics workforce. Over 95% of RISS alumni state that the RISS experience had a profound impact on their future education and career trajectory. **The RISS community successfully helps to launch students into robotics.**

The RISS community has hosted research experiences for students from over 75 home countries plus cities and towns across the United States. 2022 scholars' home countries included the United States, China, Germany, India, Japan, Mexico, Nigeria, Poland, Puerto Rico, Saudi Arabia, and Vietnam. **The RISS program strives to be diverse, global, and inclusive.**

The RISS 2022 Cohort:

- 47 scholars
- 40 home universities
- 11 countries of citizenship

The RISS 2022 program was a homecoming that brought students, mentors, and the community back to campus for our first in-person program in more than two years. The 2022 cohort was selected from an applicant pool of over 700 applications from more than 40 countries and over 300 institutions worldwide.

RI community engagement with RI Summer Scholars lasts for years. Core to the RISS values is ongoing support and connections. One concrete example is the opportunity to return for additional mentored research experiences. The following cohort members were returning RISS alumni: Jacob Adkins, Shaden Alshammari, Rayna Hata, Chigozie Ofodike, Ernest Propek, Conner Pulling, Grace Su, and Renos Zabounidis.

RISS scales access. Two new initiatives were launched in 2022 to scale access and engagement in robotics: RISS Robolaunch and the Pennsylvania Robotics Scholars. Our continued collaboration with the United Air Force Academy and the German Academic Exchange Service (DAAD) connects domestic and transatlantic robotics research communities. These partnerships were strengthened in 2022. The RI Community also began to explore connections with scholars in Poland.

The RISS Robolaunch virtual seminar series reached thousands of participants worldwide. The RISS RoboLaunch series featured smart, digestible, and inspirational robotics briefings from scientists, entrepreneurs, and educators. The talks and workshops brought "big ideas" in robotics, automation, and artificial intelligence to thousands of participants worldwide. *We thank the TC Energy Foundation for their generous support that made RISS Robolaunch possible!*

Robo Launch
A Robotics Outreach Initiative

The Pennsylvania Robotics Scholars pilot connected five PA undergraduate students with faculty mentors and projects, scholarship funding, and professional development. We would like to thank all of our partners that joined us in a campaign to increase awareness and open doors to funded research opportunities — with special thanks to CMU's School of Computer Science and the Robotics Institute for providing the funding & support that makes this work possible and to the Pennsylvania Department of Education (PDE), Dr. Tanya Garcia, and Judd Pittman for their work to increase access and awareness of STEM opportunities across the state.

The German Academic Exchange Service (DAAD) collaboration opened RI doors to German scholars. The DAAD RISE Worldwide scholarship program enables German undergraduate students to conduct scientific research around the world. Immersing in global research and lived experience enables these emerging scholars to build connections and human understanding vital to shaping a more equitable and sustainable future. Three students were awarded DAAD RISE scholarships to join Carnegie Mellon's Robotics Institute Summer Scholar's (RISS) 2022 cohort. Rachel Burcin was awarded a 2021 Germany Today fellowship to meet with leaders & innovators from German higher education institutions. In 2022, scholars met with delegations from the Ruhr Region and Baden-Württemberg (Germany) to discuss global science and technology networks.

RISS explores science and technology policy with leaders. Scholars explore economic development, STEM education pathways, and policy through facilitated visits with political, industry, and community leaders. Scholars participated in policy dialogue with German industry and government leaders and Pennsylvania education and policymakers.

Carnegie Mellon, our sponsors, mentors, alumni, and partners are committed to building more inclusive pathways in robotics. We extend our sincere thanks to all the heroes that make this happen!

With gratitude,

John & Rachel

Dr. John M. Dolan
Director of RISS Program
& Principal Systems Scientist
jdolan@andrew.cmu.edu

Ms. Rachel Burcin
Co-Director of RISS Program &
Global Programs Manager
rachel@cmu.edu

Robotics Outreach

Pennsylvania Scholars' Pilot

Anisa Callis

Joel Julin

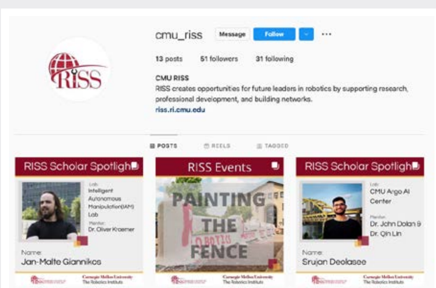
Kaylie Barber

Kylie Barber

William Scott

Thank you Judd Pittman & Pennsylvania Department of Education, for partnering to launch the PA Scholars Pilot.

Online Outreach



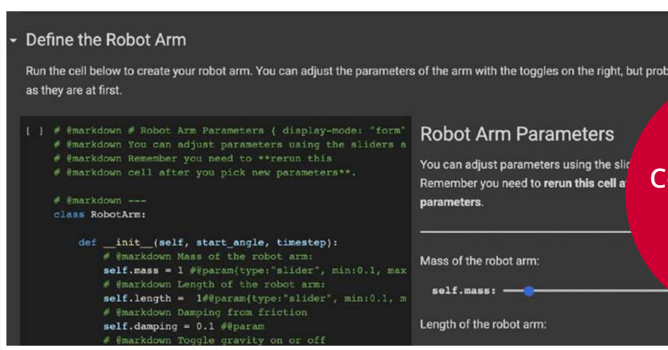
RISS Instagram Launched!
@cmu_riss

RISS Twitter:
@cmu_riss

RISS LinkedIn:
CMU Robotics Institute Summer Scholars Program

Outreach led by Scholar William Scott for the Gelfand Center

Rapid Prototyping Workshop with Dr. Jordi Albo



Feedback Control and PID RoboLaunch Workshop

STEM Education Policy Dialogues

Meeting with County Commissioner Chip Abramovic

The RISS RoboLaunch Initiative is a robotics outreach & broadening participation initiative developed by the RISS community.

riss.ri.cmu.edu/robolaunch

13,000+
Viewers

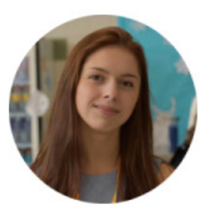
A special thank you to the RoboLaunch sponsor: TC Energy Foundation

RoboLaunch Organizers

Rachel Burcin
RISS Co-director & Global Programs Manager
Carnegie Mellon University

John M. Dolan, PhD
RISS Director & Principal Systems Scientist
Carnegie Mellon University

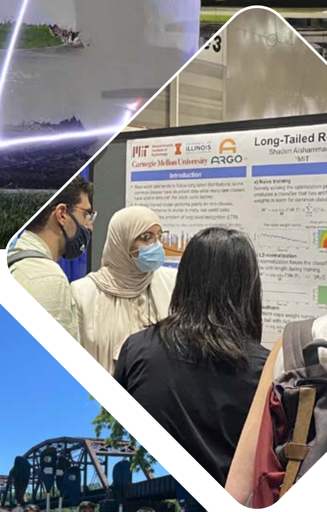
Nikhil Varma Keetha
Masters Student
Carnegie Mellon University



Taylor Koda
RISS Intern
Carnegie Mellon University

Daniel Young
Media Intern
Slippery Rock University

Brady Moon
PhD Student
Carnegie Mellon University



From the Scholars...

The Robotics Institute Summer Scholars Working Papers Journal is an annual publication of the Robotics Institute's Summer Scholars Program at Carnegie Mellon University. The journal allows the RI Summer Scholars to communicate their research contributions and experiences. The journal explores a variety of robotics domains, including localization, mapping, computer vision, motion planning, controls, haptics, aerial systems, medical robotics, multi-agent systems, machine learning, and reinforcement learning.

The RISS Scholar Journal Team would like to thank fellow scholars for guiding the peer-review process and enhancing the quality of the papers. Additionally, the scholars would like to acknowledge all the support that Carnegie Mellon, especially the CMU Center for Student Academic Success Center, provided through its writing, professional development, and research programming.

The scholars would like to thank the mentors for their invaluable guidance and feedback throughout the program. The scholars would also like to acknowledge all the outstanding speakers who joined us for RoboLaunch and the RISS External Marketing Team for their organization and support of RoboLaunch and Social Media platforms.

Finally, the cohort would like to thank RISS co-directors Ms. Rachel Burcin and Dr. John M Dolan, who have tirelessly put their time and effort into making this program possible in person for the first time in two years. The RISS experience was only possible with their hard work, coordination, advice, and enthusiasm. We are also thankful to the RISS community for their contributions and support of the program.

— The RISS Scholar Journal Team

RISS | ROBOTICS INSTITUTE SUMMER SCHOLARS

Thank You

Program Sponsors & Partners

NSF

We gratefully acknowledge the support of the National Science Foundation IIS Div of Information & Intelligent Systems through the Research Experience for Undergraduates (REU) program (Grant # 1950811).

Carnegie Mellon University
The Robotics Institute

Carnegie Mellon University
School of Computer Science

AF
UNITED STATES
AIR FORCE ACADEMY

Mobility21
A USDOT NATIONAL
UNIVERSITY TRANSPORTATION CENTER

DAAD
Deutscher Akademischer Austauschdienst
German Academic Exchange Service

KGSP
برنامج جامعة الملك عبدالله
للعلوم والتقنية للطلبة الموهوبين
KAUST GIFTED
STUDENT PROGRAM

AIRS
SHENZHEN INSTITUTE
of ARTIFICIAL INTELLIGENCE AND ROBOTICS for SOCIETY
深圳市人工智能与机器人研究院

上海科技大学
ShanghaiTech University

Carnegie Mellon University
Student Academic Success Center

Carnegie Mellon University
University Libraries

Carnegie Mellon University
Mechanical Engineering

Working Papers

Improving Conflict Detection in Scalable Multi-Robot Motion Planning

Abdel Zaro[1], Ardalan Tajbakhsh [2], and Aaron M. Johnson [2]

***Abstract*— Multi-Agent Motion Planning involves finding dynamically feasible plans for robots to reach their goal states. While scalable methods with hundreds of robots have been presented, many assumptions made limit real-world deployment. Conflict-Based Search (CBS) is an effective approach for finding optimal and complete paths; however, robot kinematics, such as turn radius, are not considered. Second, many CBS approaches assume a robot occupies a location at a single timestep rather than over an interval of possible times. Finally, current methods utilize pairwise collision detection, which is computationally expensive and often a scalability bottleneck, especially for real-world implementation. In this study, these assumptions are relaxed. To decrease the computation complexity of current CBS collision detection methods, quadtrees are utilized to limit collision checking to nearby robots. Model Predictive Control is utilized as the low-level planner of CBS to generate future robot trajectories. Experimental results show the implementation of quadtrees can decrease the total collision checking time from 4.568 seconds down to 0.046 seconds for a 100-robot system. The time savings demonstrated hold promise for scalable real-world robot motion planning.**

***Index Terms*— Multi-Robot Systems, Path Planning for Multiple Mobile Robots, Collision Avoidance, Conflict-Based Search**

I. INTRODUCTION

Multi-Agent Motion Planning (MAMP) simultaneously directs many robots to their goal positions while respecting dynamics and avoiding collisions with other robots and the environment, as illustrated in Fig. 1. Solving this problem enables applications including warehouse fulfillment, environmental sampling, disaster recovery, and material handling. These all require many robots to perform numerous tasks while interacting with each other and the environment. Moreover, these robots need to plan their motions in real-time in order to react to any unexpected changes in the environment.

While existing MAMP solutions such as Conflict-Based Search (CBS) can generate conflict-free trajectories for many robots [1], many assumptions are made that limit their ability to be implemented in real-world settings. Robot kinematics are often not taken into account. Further, the environment, time, and the action space are often discretized and unit action duration at each timestep is assumed. Current methods handle robot-robot conflicts as a rigid location-time pair, which can be inaccurate considering execution imperfections such as localization drift, wheel slippage, and tracking error. Most of these assumptions will likely be violated during execution when robots turn, interact closely with one another, or accelerate. In this paper, these assumptions are not made. Robot kinematics are taken into account and robot actions are carried out in continuous time to account for execution imperfections.

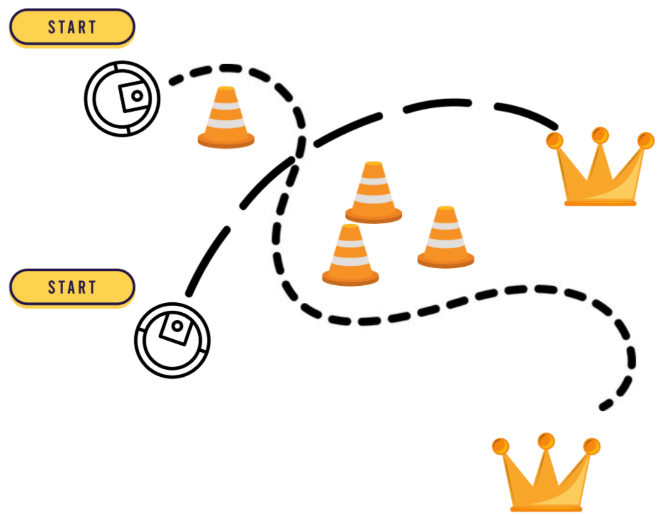


Fig. 1: Illustration of multiple robot plans that avoid other robots and environmental obstacles.

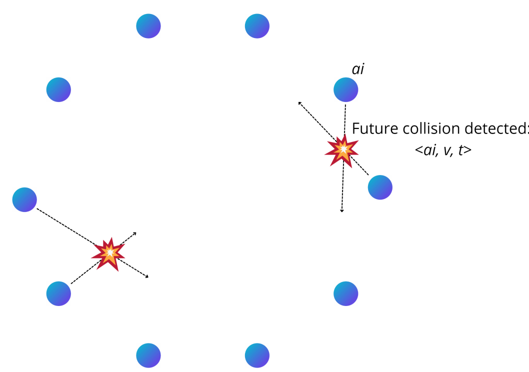


Fig. 2: Visualization of a collision between robot ai and another robot at location v and timestep t.

Although variations of CBS that relax some of these assumptions have been studied, there has not been a focus on improving the efficiency of the collision detection part of the algorithm. Current methods involve pairwise collision detection, which does not scale well with many robots. In this paper, a modified CBS algorithm that addresses the scal-

[1]Abdel Zaro is with the Robotics Institute Summer Scholars Program at Carnegie Mellon University, Pittsburgh, PA 15213, USA and also with the Department of Mechanical Engineering at The University of California, Berkeley, CA 94720, USA abdelzaro@berkeley.edu

[2]Ardalan Tajbakhsh and A. M. Johnson are with the Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA atajbakh, amj1@andrew.cmu.edu

ability issue is proposed. Model Predictive Control (MPC) is used as CBS's low-level planner, allowing for planning over a receding horizon. Additionally, MPC provides more natural incorporation of constraints than other CBS low-level planners such as A*. The main contribution of this paper is using quadtrees to improve the efficiency of CBS collision checking for large-scale problems.

II. RELATED WORKS

A. MAMP Solvers

Centralized MAMP approaches have a single CPU controlling all agents and they fall under two major categories, decoupled and coupled. In the decoupled approach, motions are planned for each individual robot separately and conflicts are resolved on-demand. With the coupled approaches, the problem is solved jointly for all robots at the same time. Decoupled approaches are often faster than the coupled ones but do not provide the same optimality or completeness guarantees.

Conflict-Based Search (CBS) is a popular centralized two-level optimal search algorithm. In the low-level portion of the algorithm, individual plans are generated for each robot separately [1]. Various decoupled algorithms can be used as the low-level planner. The high-level planner detects conflicts between the paths generated by the low-level planner. A state in the search space is described by a location and timestep (v, t). When a collision is detected, as shown in Fig. 2, a constraint is added to prevent the robot, denoted by ai, from being at location v at timestep t, represented as $<ai, v, t>$. A binary constraint tree is formed. Each node has constraints, a solution, and the total cost. Constraints of each child node of the tree are inherited from the parent node. Additional constraints may be added during the progression of the search to prevent collisions between robots. Within the solution of the node, each agent has a set of paths generated by the low-level planner, all of which satisfy the constraints at that time step. It's important to note that if the constraints are not sufficient at that time step, meaning that the current constraints do not prevent all the collisions, additional constraints must be added. The total cost of the node is the summation of all the path costs. The search is ended once a solution that only contains collision-free paths is found.

Algorithm 1 shows the basic CBS algorithm proposed by Sharon [1]. The low-level planner is used to find the shortest path for each robot without considering the other robots (line 2). The constraints, solutions and solution cost describe a node of the constraint tree and are passed into the open list (OPEN) (line 4). Each node is iterated over and checked for whether the solution it holds would result in a collision-free path (line 8). If a collision-free solution is found, the solution is returned (line 9). Otherwise, for every robot, ai, involved in the collision, new constraints are added to prevent that robot from being at location v at time step t (line 11). With the addition of the new constraint, the low-level planner is called to generate a new path for the robot at hand (line 15).

Algorithm 1: high-level of CBS

```
   Input: MAPF instance
1  R.constraints = ∅
2  R.solution = find individual paths using the
   low-level()
3  R.cost = SIC(R.solution)
4  insert R to OPEN
5  while OPEN not empty do
6      P ← best node from OPEN // lowest solution cost
7      Validate the paths in P until a conflict occurs.
8      if P has no conflict then
9          return P.solution // P is goal
10     C ← first conflict (a_i, a_j, v, t) in P
11     foreach agent a_i in C do
12         A ← new node
13         A.constriants ← P.constriants + (a_i, s, t)
14         A.solution ← P.solution.
15         Update A.solution by invoking low-level(a_i)
16         A.cost = SIC(A.solution)
17         Insert A to OPEN
```

Finally, this node is added to OPEN again to be iterated over once again.

B. Collision detection

One of the most time-consuming parts of CBS is collision detection. This is mainly due to the pairwise-collision checking method, which involves computing the distance between every pair of robots at each time step [2], as shown in Fig. 3. A collision is detected when the shapes of the two robots overlap. If there are n robots, then pairwise-collision checking operates in O($n \times (n$-$1)/2$) time [3], which has to be repeated at each timestep.

Multiple methods have been proposed to overcome the computational complexity of pairwise collision checking. Velocity obstacle is a method in which robots adjust their velocities to avoid other robots [4]. This is done by constructing a region called a velocity obstacle for each robot, which depicts all the velocities that would result in a collision. As long as the robot selects a velocity outside the velocity obstacle, the collision is prevented.

Another method proposed to speed up the collision checking process involves safety certificate regions [5]. These regions are formed by measuring the distance from the robot to the closest obstacle and then forming a circle around the robot with a radius equal to that distance. If the robot is within the safety certificate, then a collision check can be skipped.

These methods still require distance computations between robot pairs at each time step and the safety certificate method does not account for dynamic obstacles, such as other robots. In order to address these issues, quadtrees are utilized in this paper. The high-level idea of quadtrees is to partition the area the robots occupy into smaller regions. These regions can be described by a tree data structure that provides a simple

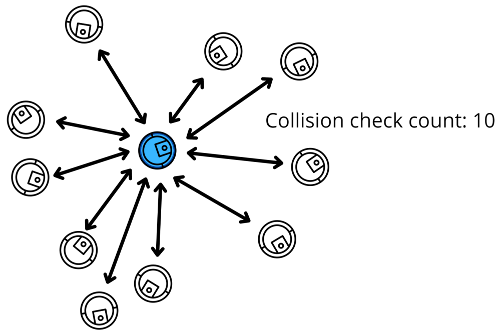


Fig. 3: Illustration of pairwise collision checking for a single robot. Ten collision checks must be done for each robot in the system to detect conflicts between every robot pair.

method for identifying which robots are near each other. This information can be used to restrict collision checking to only robots that are within proximity. The algorithm starts by drawing a bounding box around all the robots. Let m be the maximum number of robots allowed to be contained within the bounding box. In this paper, this value is set to two robots. When there are more than m occupying a space, the quadtree algorithm creates four new quadrants, as illustrated in Fig. 4.

Fig. 4 a.) shows two robots within the same bounding box. Since m was set to two, no new quadrants are formed. Fig. 4 b.) illustrates what happens when a third robot, C, is added. The space is partitioned into four equal-sized quadrants. As the bounding boxes are split, a tree is constructed, as shown on the left. The outer bounding box is called the parent node and the four quadrants within it are called the child nodes. Each robot is placed into the child node of the tree that corresponds to the quadrant that they occupy. For example, robots B and C are in the third quadrant and thus are placed in the corresponding child nodes of the tree. Whenever the number of robots occupying a bounding box exceeds m, the quadrant they occupy is yet again split into four smaller quadrants, and a new level of the tree is formed. Fig. 4 c.) and d.) demonstrate the tree expansion as even more robots are added.

The main advantage of quadtrees is limiting the number of collision checks to only the nearby robots. The reduced number of collision checks allows quadtrees to operate in O$(nlog(n))$ time [6].

III. APPROACH

A. Overview

Each robot position (x, y) is represented as a circle with a radius of 3. Tests with 10, 50, and 100 robots arranged in a circle with a radius of 20, 50, and 75, respectively, are conducted. Robots were placed in a 500x800 pixel area. Each robot swapped positions with a robot on the other end of the circle as shown in Fig. 5. Model Predictive Control (MPC) is used to predict each robot's current and future trajectories at every timestep of 0.2 seconds. This is done to enable early collision detection. Experiments were done in

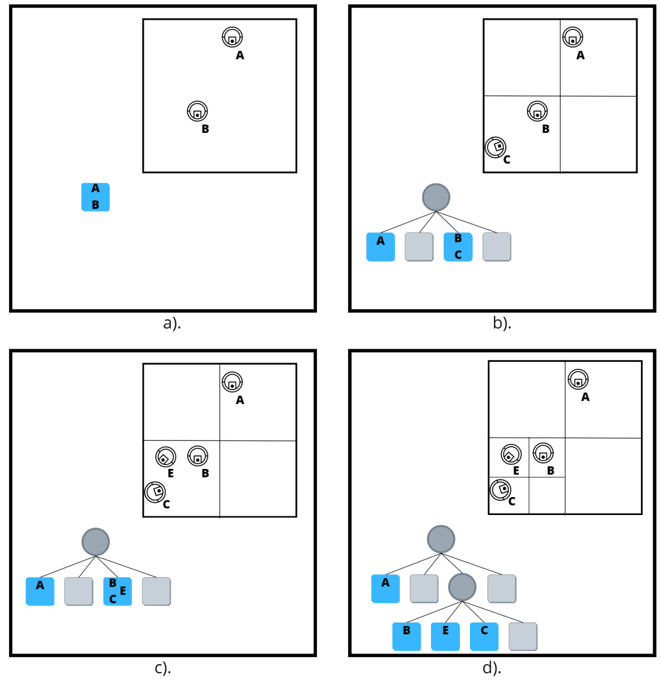


Fig. 4: a). Only two robots are within the bounding box, which is equal to m, so splitting the box is not required. b). A third robot is added to the bounding box and the total number of robots exceeds m, so the space must be split into quadrants. c). An additional robot, C, is added. d). The space must again be split.

Python on a laptop with an Intel® Core™ i7-7700HQ with 16 GB of RAM. The Euclidean distance was computed for every collision check between robots.

A collision between robot ai and aj occurs when the distance between them is less than the sum of their radii. It is assumed that all the robots are the same shape and size. Collisions are checked over the entire MPC horizon, which holds the predicted trajectory information of the robot up to 5 timesteps in the future.

B. Implementation

A naive pairwise collision checking approach is taken as the baseline. At every timestep, each robot does a collision check with all the other robots. It operates in O$(n \times (n\text{-}1)/2)$ time; thus, it scales quadratically with the addition of robots.

Next, a quadtree is tested to minimize the number of collision checks between every robot. At every timestep, collision checks are only done between robots within the same bounding box, which limits collision checks to only nearby robots.

IV. RESULTS

As shown in Table 1, in a 100-robot system, the use of quadtrees decreased the total collision checking time from 4.568 seconds down to .046 seconds. This is because quadtrees enable the algorithm to only do collision checks on robots that are nearby and within the same bounding box. The benefit of quadtrees becomes even more apparent as the

< TABLE OF CONTENTS

TABLE I: Total collision checking time and number of collision checks

# Robots	Pairwise collision checking time (sec)	Quadtree collision checking time (sec)	# Pairwise collision checks	# Quadtree collision checks
10	0.048	0.008	3,780	316
50	1.279	0.031	139,650	1,882
100	4.568	0.046	405,900	2,349

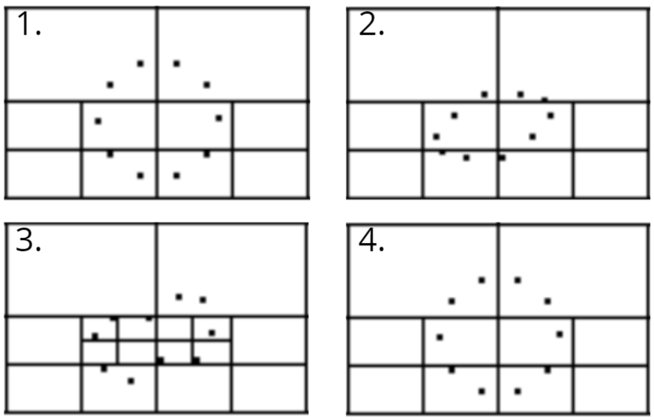


Fig. 5: Ten robots swapping positions with a quadtree implemented.

number of robots increases. Thus, it contributes to solving the scaling issue of multi-robot systems. Conversely, pairwise collision checking faces a quadratically increase in collision checks when more robots are added. This is because every robot has to be checked against all the other robots, which causes pairwise collision detection to operate in O($n \times (n$-$1)/2$) time.

V. CONCLUSIONS

This paper implemented quadtrees to improve the scalability of multi-robot motion planning algorithms such as Conflict Based Search. It was demonstrated that quadtrees could decrease the number of collision checks by 400,000 for a 100-robot system. This improvement was due to organizing robot locations in a quadtree which improves the scaling of collision detection by limiting checks to only nearby robots. Future studies can further investigate methods to improve collision checking, such as skipping collision checks between timesteps.

ACKNOWLEDGMENT

I would like to thank Rachel Burcin and Dr. John Dolan for organizing the Robotics Institute Summer Scholars (RISS) program. I would also like to thank my mentor Ardalan Tajbakhsh, for all his help and support. Finally, I want to thank Dr. Aaron Johnson for giving valuable feedback throughout this project. This research would not have been possible without the National Science Foundation under Grant No. 1659774.

REFERENCES

[1] G. Sharon, R. Stern, A. Felner, and N. Sturtevant, "Meta-agent conflict-based search for optimal multi-agent path finding," in *International symposium on combinatorial search*, vol. 3, no. 1, 2012.

[2] A. Andreychuk, K. Yakovlev, P. Surynek, D. Atzmon, and R. Stern, "Multi-agent pathfinding with continuous time," *Artificial Intelligence*, vol. 305, p. 103662, 2022.

[3] M. H. J. Saldanha and P. S. L. d. Souza, "High performance algorithms for counting collisions and pairwise interactions," in *International Conference on Computational Science*. Springer, 2019, pp. 182–196.

[4] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The international journal of robotics research*, vol. 17, no. 7, pp. 760–772, 1998.

[5] J. Bialkowski, M. Otte, S. Karaman, and E. Frazzoli, "Efficient collision checking in sampling-based motion planning via safety certificates," *The International Journal of Robotics Research*, vol. 35, no. 7, pp. 767–796, 2016.

[6] R. A. Finkel and J. L. Bentley, "Quad trees a data structure for retrieval on composite keys," *Acta informatica*, vol. 4, no. 1, pp. 1–9, 1974.

[7] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[8] F. Wireland, D. Carnes, and G. Schultz, "Using quad trees for parallelizing conflict detection in a sequential simulation," in *Proceedings 15th Workshop on Parallel and Distributed Simulation*. IEEE, 2001, pp. 117–123.

A tensioning mechanism for an open source educational jacquard loom

Ana P. Garcia-Alonzo, Samantha Speer, James McCann and Melisa Orta Martinez

***Abstract*— This paper introduces the design of a novel three stage tensioning mechanism for an open source, automated, Jacquard table loom designed specifically for educational applications that can be assembled by students. Our aim is to motivate and concretize learning in math, computational thinking, and robotics through craft practices and hands-on learning activities associated with weaving. To be able to relate math concepts to weaving patterns, it is important for students to be able to create distinguishable high quality patterns. To produce such high quality cloth, uniform tension in the warp threads needs to be maintained throughout the weaving process. In this work we propose the design of a tensioning mechanism that allows for individual thread tension force adjustment and measurement during the weaving process. This mechanism provides a unique approach in the application of spring, friction, and normal forces in order to maintain uniform tension throughout the cloth.**

***Index Terms*— weaving, yarn tension, interactive learning, mathematics, hands-on education**

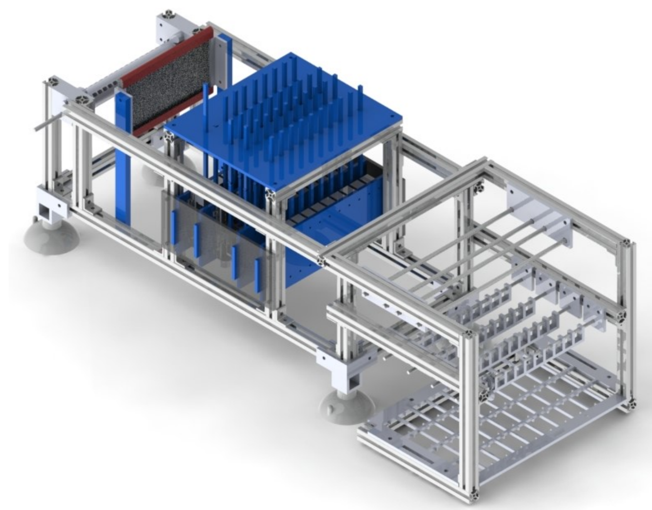


Fig. 1. Robotic loom, with 40 individually actuated threads, designed to be assembled as a kit by teams of students to encourage collaborative learning. Made from readily available components and 3D printed parts. The combined tensioning mechanism allows users to tension each thread individually, as well as modify the force of the tension as they weave. The loom is controlled through an Arduino Mega board and programmed in the Arduino platform.

I. INTRODUCTION

A. Motivation

Students often encounter difficulty in understanding abstract concepts in STEM subjects such as physics and mathematics. Mulwa [1] proposed that part of the problem is the lack of grounding of the abstract terminology. To address this problem, it has been proposed to modify the teaching methods of these subjects by increasing the use of technology, hands-on activities, and use of real world objects and examples [2]–[5]. Successful examples of such technologies that aid in the understanding of theoretical concepts are Hapkit [5] and Phogo [6]. These robotics projects have become accessible to students through 3D printing technology and a wide research community sharing a common goal of making education more accessible.

The nature of the weaving process and design of weaving structure which can be interpreted as matrix operations and analyzed through computational thinking paradigms, provides a unique opportunity to ground abstract mathematical concepts in art. In order to connect these concepts together we have designed an open source Jacquard loom kit meant to be assembled by groups of 4 students and programmed using matrix operations and the Arduino programming language. In this paper we introduce the design and analysis of the tensioning mechanism which is crucial to the functioning of the loom and integrity of the produced cloth.

B. Background

Weaving is a fabrication technique that has been around for thousands of years. Different cultures have developed their own processes and techniques, incorporating a wide variety of materials. The foundation of weaving lies in the manipulation of warp threads, which run vertically, and weft threads, which run horizontally (Figure 2). Woven fabrics have a wide range of usages, each requiring different looms to fit the complexity of the weave. Contemporary industrial Jacquard looms, which allow for complex fabrics, are very expensive for individual use. Their price is proportional to the size and fineness of the cloth they can weave. In e-commerce companies, such as Alibaba, one can buy a computerized industrial GINYI Jacquard loom from a starting price of \$7,500 USD or a WGT16 high-speed electronic Jacquard weaving loom for \$60,000 USD. On the other hand, simpler commercial hand or table looms do not support complex patterns and often require a previous study on weaving and specific techniques [7]. In this project we consider the trade-offs in cost, performance, and accessibility to create a lower cost Jacquard robotic loom.

There have been other low-cost robotic Jacquard looms which seek to innovate on specific parts of the loom. One example is a computer controlled loom designed by Schaefer [8], which lifts or lowers the yarn by shifting hooks through cams that sit on an automated rotating square shaft which can be controlled by a cellphone. This loom was designed as a Do-It-Yourself (DIY) project and made from 3D printed materials and simple electronic components. Other

< TABLE OF CONTENTS

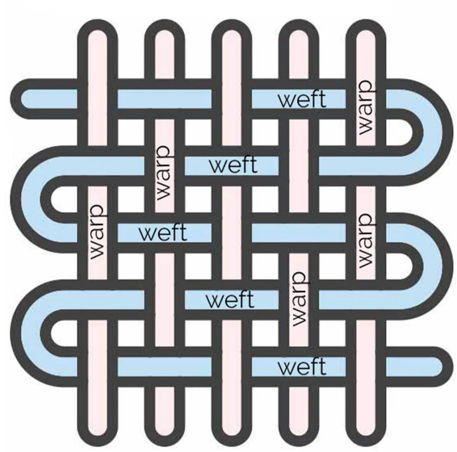


Fig. 2. Visual representation of warp & weft packages. The vertical strings correspond to the wrap, which are held by the tensioning system of the loom, and alternate their position with respect to the weft. In contrast, the weft corresponds to the horizontal rows which are woven manually by the user.

small mechanical looms have been created from popular components such as LEGO, which provide a wide range of mechanical and electronic components that can emulate up to a fully automated Jacquard loom [9]. Albaugh et al. [7] created a 3D printed low-cost Jacquard Loom that offers fully computational pattering, while maintaining flexibility of hand weaving, and allows for multiple levels of complexity. Said design aims to serve artisans and researchers by facilitating a rapid prototyping and accessible computational interface for skill-building and creative development.

Our loom aims to combine a creative, low-cost design, such as the ones created by Kurt and Nichols, with the computational and mechanical flexibility that Albaugh's design provides in order to facilitate the integration of weaving and computational thinking.

II. DESIGN PROCESS

Our main goal in designing the tensioning mechanism for RoboLoom, was to design a mechanism that allowed for each thread to be individually tensioned at a load between 50 and 200 g and was easy for students to assemble and adjust.

Figures 3, 4, and 5 present the evolution of our mechanism. Figures 6 and 7 show the final design. In the following sections, we describe the design goals and design evolution for this mechanism.

A. Design Goals

The aim of this open source Jacquard loom is to allow undergraduate students to fully assemble, interact and skillfully apply the theoretical knowledge acquired during an interdisciplinary course combining concepts in mathematics, robotics, computational thinking and weaving. Therefore, the loom should be robust, intuitive, and easy to assemble and use. It should also allow for corrections, since mistakes are bound to be frequent amongst students. Especially those with no prior experience in weaving.

One of the biggest challenges in designing a loom is designing a tensioning mechanism which maintains uniform tension amongst threads whilst allowing for adjustments as one weaves. In weaving, having sufficient -between 50 and 200 grams of tension- and consistent tension in the yarn is crucial for fabric integrity. In order to achieve this, we propose a design which allows for individual manipulation and tensioning of the warp threads. This mechanism also aims to reduce the points of friction which each wrap may encounter along the loom, avoiding rupture or inappropriate pull caused by the movement of the heddles. Finally, taking in consideration that the desired application for this loom is for it to be used within introductory courses for robotics, mechatronics and computer science, the assembly should be able to be completed in a reasonable amount of time by teams of 4 students.

B. Manufacturing

In order for RoboLoom to be an accessible educational tool, students, as well as educators, should be able to obtain, modify, and replicate the parts as easy as possible. Our loom is composed of 60% readily available materials, 20% laser-cut acrylic, and a remaining 20% of 3D printed parts.

In order to make the assembly process as unambiguous as possible, the 3D modeled parts aim to reduce the mechanism to a minimal number of parts and lower the inconsistencies while printing. The material for each of the printed components was carefully selected depending on its interaction with the yarn threads. For the compressing system found in the tensioning discs, we decided to use stereo lithography (SLA) 3D printing because of the highly-precise layer to layer quality to avoid scratchy edges, making it easier for the thread to slide through despite all the force that is being applied, thus reducing possible points of rupture. For other components whose function is mainly as storage, the chosen material was poly lactic acid filament (PLA), commonly used in 3D nozzle extruding printing machines which are more readily available and inexpensive.

C. Design Evolution

The initial designs aimed to increase the manipulation of the individual warp threads within a 40 thread loom. The system found in traditional table looms allows for control over sets of threads but not individual threads. Control over each single thread allows for more complex, flexible and creative patterns.

The first design (Figure 4) was a fixed tension-system with a bobbin and the tensioning spring were held inside a case. The force on the tensioning discs was set by the distance between the lid and the base. Having the bobbin fixed in the base, so it had to be spooled inside the structure, proved to be impractical and hard to assemble.

As the design evolved, we sought to make a mechanism where the tension could be readjusted as needed and the bobbin was easier to wind. This then resulted in the design shown in Figure 5, where the spool was a rolling component, and the tension on the discs was adjustable with a spring

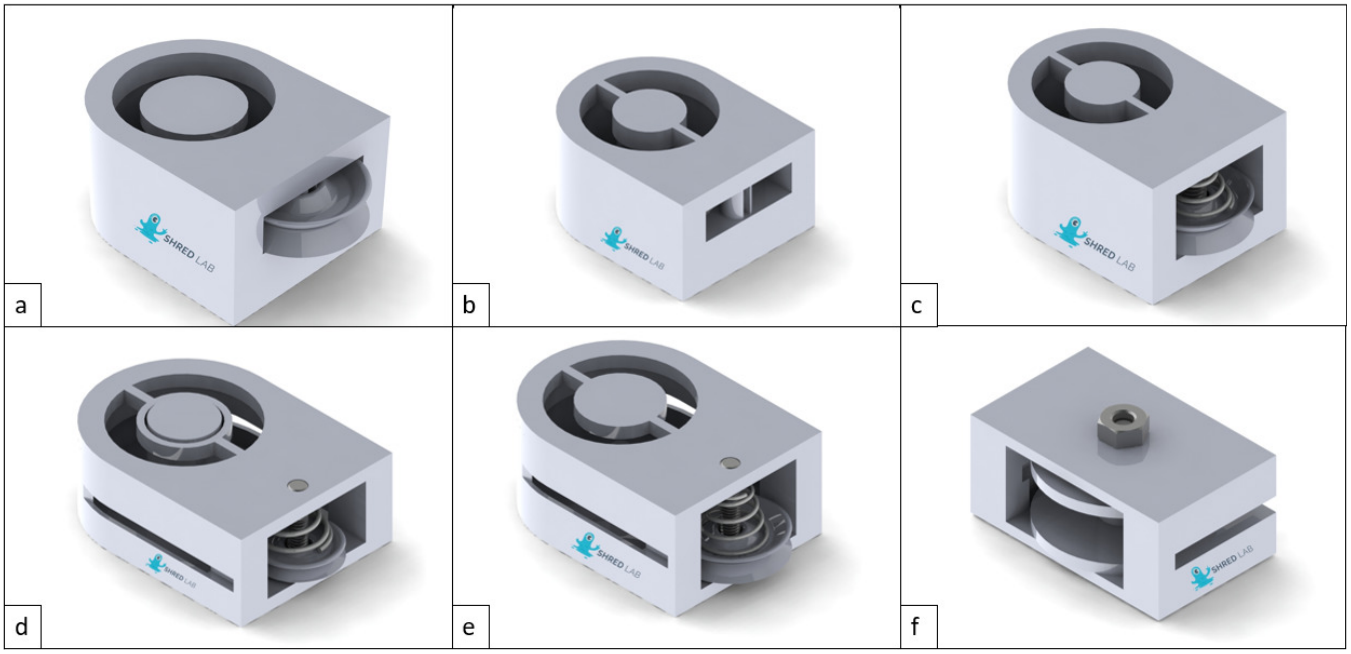


Fig. 3. Design evolution of tensioning system, a) first design with no grip over bobbin, b) second design with fixed tension, no spring, c) third design with spring, d) forth design with rolling bobbin and small spring, e) fifth design suitable for vertical assembly and adjustable tension, f) bobbin holder separated form tensioning system.

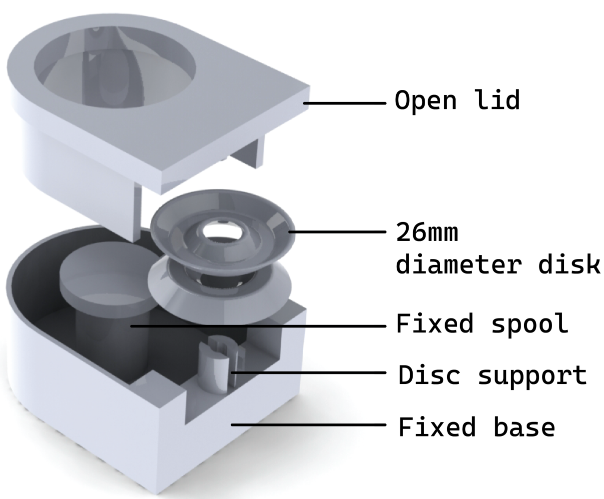


Fig. 4. Initial tensioning mechanism design. The tension is fixed in the discs by an established distance between the lid and the base. The spool was also fixed in the base, which made it complicated to weave the thread into it.

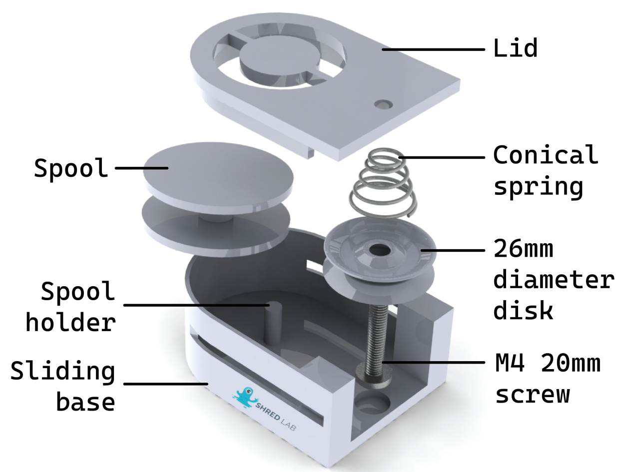


Fig. 5. Evolving tensioning mechanism design. The space between the lid and base is bigger to allow adjustment of the spring with a nut, increasing the tension between the discs. The discs are held by a 20 mm screw that fits into the lid, bounding the structure together. The spool is a separate component that holds around 24 ft (7.32 m) of yarn. The side slots allow the assembly of the entire case into a secondary structure that sits in the back of the loom.

and nut. The rolling spool allows for easier movement of the thread as it leaves the case. The spring facilitated the readjustment of the tension, however, it proved to be imprecise when the exact position of the nut had to be replicated for each of 40 threads.

The final tensioning mechanism design was split into two parts since we found it was impractical to hold the thread so close to the tensioning mechanism if there was any need to pull back the thread and readjust. The first part (Figure 6), is the storage system, and includes an freely-spinning bobbin and an arrow to guide users in threading. The second part (Figure 7), is the tensioning system, which resolves the issue of having to calibrate 40 systems by combining a tensioning spring with a fixed-spacing carrier assembly. To save on assembly time and materials, we built a 10-tensioner module using a 3D printed a spacing rod. Only four of these modules are needed for the 40-thread loom.

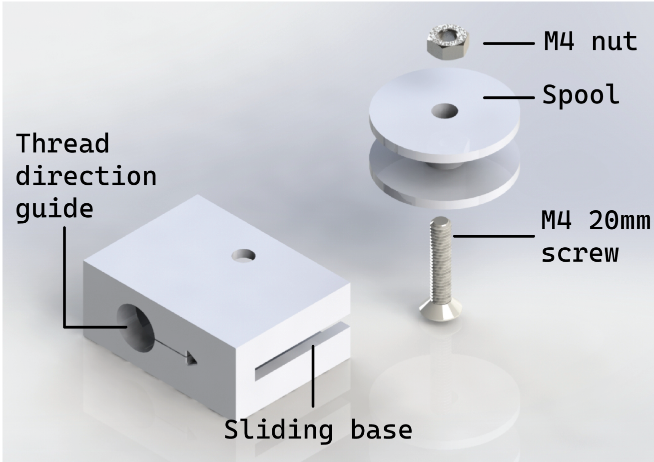


Fig. 6. Storage system for the final tensioning mechanism. A spool that holds around 24 ft (7.32 m) of yarn is held by a sliding case that rests below the tensioning system. The arrow slot helps orient the thread in the right direction during the assembly process.

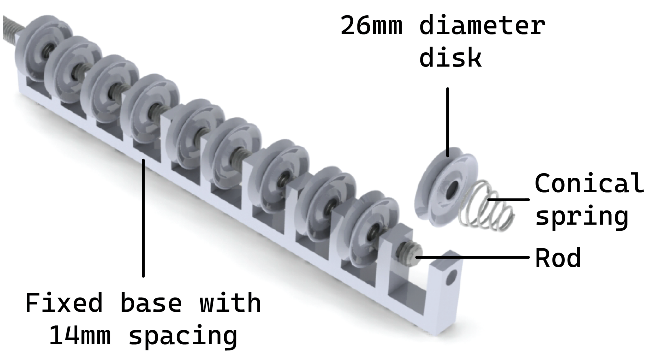


Fig. 7. Final tensioning mechanism design. A structure that holds a fixed 14 mm spacing allows for consistent tension between 10 threads at the time. The springs are compressed in this fixed spacing against the discs, which are all held by the same stainless steel rod. The amount of threads is equivalent to one heddle frame, which holds these same threads, facilitating the organization of the yarn through the loom.

D. Tensioning mechanisms

The tensioning mechanism was designed so it could be adjusted and hold the tension it was adjusted to throughout the weaving process. We focused on designing a mechanism that was passive (with no active actuation components) in order to minimize cost. Here we describe the tensioning paradigms which informed our design.

1) Additive tensioning systems: The simplest way of applying tension to yarn is to rub it against another surface [10]. The magnitude of the tension depends on the force with which the thread is held against the disk and the coefficient of friction (μ_1) of the resin the disk is made out of. This is known as an additive tensioning system, and is usually made of two solid stainless steel discs compressed by a spring (Figure 8). The force applied by the system is given by Equation 1:

$$F = \mu_1 N + \mu_1 N = 2\mu_1 N \tag{1}$$

If a yarn is fed through this system with an initial tension T$_i$, then its output tension, T$_f$, is determined by Equation 2:

$$T_f = T_i + 2\mu_1 N \tag{2}$$

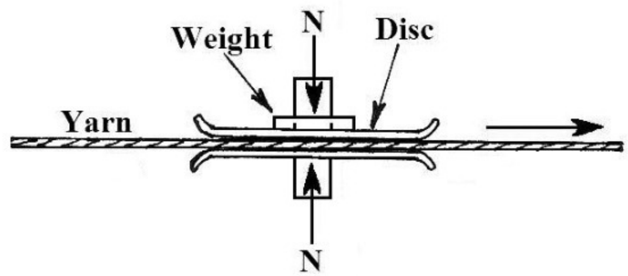


Fig. 8. An additive tensioning mechanism with two discs. The discs compress the yarn as it passes through them, resulting in an increase in its final tension. Figure from [10].

2) Multiplicative tensioning systems: Another traditional method is known as a multiplicative system, Figure 9, in which the yarn passes around a fixed, curved shaft [10]. The output tension, T$_f$, on the thread is now determined by the angle of the curvature and the coefficient friction of the shaft (μ_2), as per Equation 3:

$$T_f = T_i e^{\mu_2 \theta} \tag{3}$$

With this method, tension can be adjusted by changing the final direction of the thread as it gets ready to go through the heddles until its final tie down. However, the final tension is also dependent on the initial tension – which must be greater than zero, else this holds no effect.

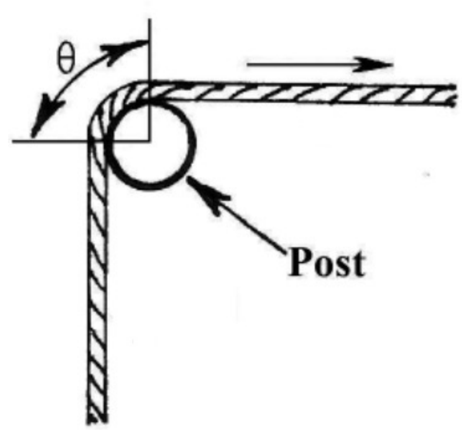


Fig. 9. A multiplicative tensioning system. The friction of the post and angle of the direction change multiply the initial tension. Figure from [10].

3) Combined tensioning system: For our design, we decided to combine both the additive and multiplicative tensioning systems. Using this combined tensioning system, we were able to create a mechanism that was easy to assemble and provides control over the tension magnitude and direction through the loom as proposed by [10]. The output tension T_f is then modeled as the addition of both systems and will depend on the normal force applied by the discs, the initial tension, the angle of the warp, and the coefficient of friction of each system, as per Equation 4:

$$T_f = T_i(1 + e^{\mu_1 \theta}) + 2\mu_2 N \tag{4}$$

E. Assembly Process

Figure 10 outlines the different sections that make up the final design of the loom as well as how the warp thread interfaces each section. Here we describe the assembly process of the tensioning mechanism shown in sections A, B, and C in Figure 10.

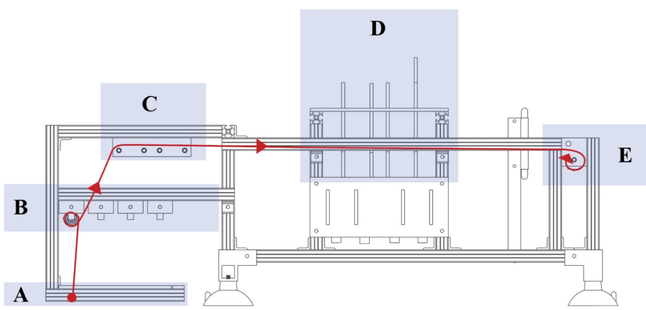


Fig. 10. Schematic of the path of the yarn across the loom (red line) and critical tension points. A) Winding & storage station: grid that holds compartments with approximately 24 ft (7.32 m) of yarn. B) Additive tensioning system. C) Multiplicative tensioning system. D) Individually actuated heddles that raise or lower the warp threads to enable the weft threads to be woven in. E) Warp beam where the warp threads are tied down.

The different sections in the loom tensioning mechanism are mounted to a C-like structure made of aluminum using laser-cut acrylic. The laser-cut acrylic pieces are designed to hold each case with a yarn bobbin at fixed distances from each other, the tensioning disks, and heddles in order to reduce possible points of friction and tangling of the threads. This C-like aluminum structure is attached to the loom's main structure using aluminum brackets and screws.

Each section of the tensioning mechanism should be assembled in order (A,B,C).

To assemble stage A an acrylic cutout with 40 square slots is first mounted into the aluminum rails. Then, each of the yarn spools are assembled into its own storage cases, shown in Figure 6, and held by a flat head screw and a nut. The thread should cross through the arrow slot so that, as it unspools, the thread remains vertical to avoid entanglement. Finally, each case should be slid into the acrylic base and secured by a second acrylic piece on top to avoid the cases from sliding out.

In order to assemble section B, ten pairs of discs and their respective springs are mounted on a rod with a spacing structure in between (Figure 7 shows the final assembled rod). The rod is longer than the spacing structure and held by acrylic mounts above stage A. The amount of tension in each warp thread is controlled by the spacing in the 3D printed spacing structure which determines the compression of each tensioning spring. Finally each warp thread, which is held in its individual case, is looped around the pair of discs directly above it.

Finally, to assemble section C, smooth aluminum rods are secured into their respective acrylic mount. Then the thread that was looped around the discs is bent over the aluminum rod directly above it, which is the same as the row number from which it left stage A. After the last step, the thread is ready to go through its respective heddle with its own motor (shown in Figure 10 as stage D) and tied around the warp beam (shown in Figure 10 as stage E).

III. Analysis

A. Methodology

In order to analyze the performance of the tensioning mechanism, we evaluated the time it took to assemble it and whether the assembly process was easy and unambiguous. Student users of this loom are not experts in the craft of weaving thus the weaving process should be user-friendly and straightforward.

We also wove three basic patterns in order to analyze how well the tension was maintained whilst weaving. Preserving the quality of the fabric allows the user to see the different patterns resulting from their mathematical designs in weaved form. This allows students to better engage both with the art of weaving and mathematics and computational thinking lessons. If the tension were not maintained, the cloth would be uneven and not of high quality, and the planned pattern would not be apparent.

B. Results

This robotic loom is designed to be assembled as a kit by a team of students of at least 4 people. A single researcher was able to assemble the final loom prototype in approximately 5 hours. From this, we estimate a team of four novice students will be able to construct the loom in less than 10 hours. We are also taking in consideration that one of the more time-consuming components, manually winding the thread around the bobbins, will not be a part of the assembly process the students will be required to do. In future work, we aim to introduce an accessory to automate the process of winding. During the assembly process, the threading of all the yarns from their bobbin, through the tensioning discs, over the back shaft, through their corresponding heddle, through the reed and tying to the wrap beam was achieved approximately in 2 hours with 2 people. This path can be observed in Figure 10. We found that the best way to facilitate the process of threading the loom is to have a clear code that helps identify which thread corresponds to each heddle and frame.

Therefore we are working on introducing a color code to avoid confusion.

In order to test the capabilities of our adjustable tensioning mechanism, we tested three weaving patterns: Plain weave, Twill weave, and Herringbone weave. These patterns are the basis for several more complicated weaving patterns. The results of these experiments are presented in Figure 11

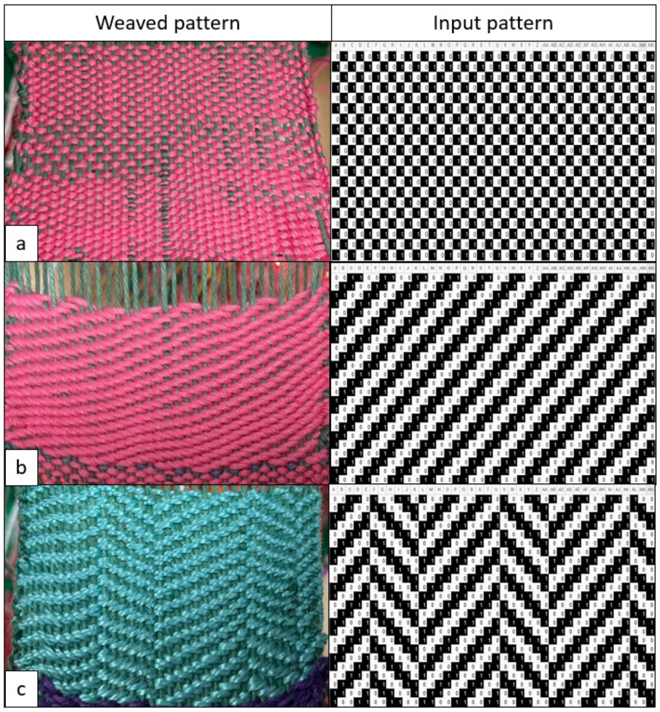


Fig. 11. Comparison between desired (right) and resulting (left) patterns. a) Plain weave. b) Twill weave. c) Herringbone weave.

1) Plain weave: A plain weave pattern is achieved by making the weft go under and over each warp thread, creating a checkerboard pattern. Figure 11.a shows a plain weave draft (right) and cloth created on RoboLoom (left). We can observe that most of the fabric holds an even distribution of the pattern. We can also see that some rows in the middle of the frame have a stronger green color, which means that at that time in the weaving process, the tension was slightly off. Thanks to the affordances of the tensioning mechanism, after realizing that the plain weave pattern showed some uneven spacing of the weft and warp, we corrected the tension as can be seen on the rest of the fabric by the even distribution of weft and warp threads.

2) Twill weave: Twill weave is characterized by a diagonal rib pattern that results from offsetting the two-over two-under structure of the weft thread interlacing. The tension in the warp threads needed to be frequently adjusted to maintain consistent tension in the cloth.

3) Herringbone weave: The Herringbone weave is composed of an alternating twill weave. We found that the tension stayed consistent, and it was possible to maintain a zigzag-like pattern, which resulted in a well-made cloth.

Another interesting experiment which allowed us to test the capabilities of our mechanism happened by accident.

Fig. 12. Correction of an individual warp thread while actively weaving: One warp thread was not weaved into the pattern for several rows due to its lack of tension. Once the weaver realized this, they adjusted the tension of that warp thread without having to untie the rest of the warp threads. The correction can be observed by the evenness of the pattern in the following rows.

Figure 12 shows a simple pattern where a warp thread was not woven into the fabric for a few rows. This happened because that particular warp thread was not tensioned properly initially. This is a common mistake in weaving, especially for beginners. Thanks to the fact that each warp thread is individually controlled and tensioned, the mistake was easily corrected by adjusting the tension of that particular warp thread without having to untie or correct any of the other warp threads in the pattern.

IV. DISCUSSION

In this paper, we presented a novel tensioning mechanism for a robotic loom that is easy to assemble and allows for dynamic adjustment of the individual tension across the warp threads. We evaluated the benefits of having individual warp thread control by weaving a set of basic patterns and measuring the consistency of the tension across the warp threads throughout the weaving process. We observed that it was easy to correct the tension in the warp threads to ensure high-quality cloth production.

We are currently working to design an undergraduate course, IDEATe: Re-Crafting Computational Thinking with Soft Technologies, which will use the robotic loom to teach introductory concepts of robotics, mathematics, weaving, and computational thinking.

A. Future work

We are working to simplify the assembly process by associating each warp thread to its corresponding heddle through color coding the cases and heddles. We also aim to continuously monitor the tension of the individual threads through an active tensioning mechanism. We also expect to find more areas of improvement after our first use case in the course.

ACKNOWLEDGMENTS

We would like to thank the Social, Haptics, Robotics, and Education Lab for providing the resources, as well as the Carnegie Mellon Textiles Lab for their assistance during the design process. Special thanks to Rachel Burcin and Dr. John M. Dolan, for making 2022 RISS possible. This loom was inspired by Lea Albaugh's work on low-cost jacquard looms. This work was supported by the School of Computer Science & Robotics Institute.

References

[1] E. C. Mulwa, "Difficulties Encountered by Students in the Learning and Usage of Mathematical Terminology: A Critical Literature Review," Journal of Education and Practice, p. 12, 2015.

[2] K.-C. Yu, K.-Y. Lin, and K.-F. Hung, "Teaching science through technology: a confluence of knowledge, design and construction," p. 6.

[3] R. Weinhandl, T. Houghton, and Z. Lavicza, "A case study on learning basic logical competencies when utilising technologies and real-world objects," Education and Information Technologies, vol. 26, no. 1, pp. 639–653, Jan. 2021. [Online]. Available: https://link.springer.com/10.1007/s10639-020-10282-5

[4] K.-Y. Lin and P. J. Williams, "Two-stage hands-on technology activity to develop preservice teachers' competency in applying science and mathematics concepts," International Journal of Technology and Design Education, vol. 27, no. 1, pp. 89–105, Mar. 2017. [Online]. Available: http://link.springer.com/10.1007/s10798-015-9340-1

[5] M. Orta Martinez, C. M. Nunez, T. Liao, T. K. Morimoto, and A. M. Okamura, "Evolution and Analysis of Hapkit: An Open-Source Haptic Device for Educational Applications," IEEE Transactions on Haptics, vol. 13, no. 2, pp. 354–367, Apr. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8878113/

[6] P. Molins-Ruano, C. Gonzalez-Sacristan, and C. Garcia-Saura, "Phogo: A low cost, free and "maker" revisit to Logo," Computers in Human Behavior, vol. 80, pp. 428–440, Mar. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0747563217305551

[7] L. Albaugh, J. McCann, L. Yao, and S. E. Hudson, "Enabling Personal Computational Handweaving with a Low-Cost Jacquard Loom," pp. 1–10, May 2021. [Online]. Available: https://dl.acm.org/doi/10.1145/3411764.3445750

[8] K. Schaefer, "Making a DIY Computer Controlled Loom," June 2019. [Online]. Available: https://retrotechjournal.com/2019/06/17/making-a-diy-computer-controlled-loom/

[9] Jerry Nicholls, "Project Weav3r: A LEGO Jacquard Loom," Jan. 2017. [Online]. Available: https://www.youtube.com/watch?v=iqlxO5RYyGU

[10] S. K. Neogi, "Role of Yarn Tension in Weaving," p. 288.

Using log data to improve RoboTutor: A Multi-Armed Bandit Approach

Abstract—**The Android tablet tutor, RoboTutor, was a million-dollar finalist in the Global Learning XPRIZE to help children without access to schooling develop literacy and numeracy. We analyzed previously collected log data from the Swahili version of RoboTutor using four criteria crucial to the effectiveness of an intelligent tutor: reliability, usability, engagement, and learning. A Multi-Arm Bandit (MAB) algorithm will use metrics for these four criteria to incrementally adjust the frequency of alternatives (arms) based on the distribution of rewards that they have achieved so far. The algorithm trades off the exploitation of the arm with the highest mean reward against exploration of other arms that have lower mean reward so far but might turn out to have higher reward. The arms are two versions of RoboTutor's curricular sequence of educational activities for which we have log data. We use this data to predict the expected increase in learning gains. The impact should be to increase total learning gains even as children are using RoboTutor in the field**

Index Terms—**Keywords: RoboTutor, Multi-Armed Bandit, Adaptive Learning**

I. Introduction

Thousands of people use technology daily to grow their skills and knowledge. RoboTutor is an intelligent tutor system (ITS) tablet first deployed in Tanzania, and is one way technology is supporting children who do not have access to school.

Fig. 1. Two students in Tanzania using RoboTutor (the tablet).

II. RoboTutor

RoboTutor and other ITS systems "aim to propose to students the activities most likely to increase their average competence level and overall knowledge components based on previous students' performances." [1] RoboTutor teaches children through adaptive learning. RoboTutor is an adaptive learning software that uses student progress to adjust the difficulty level

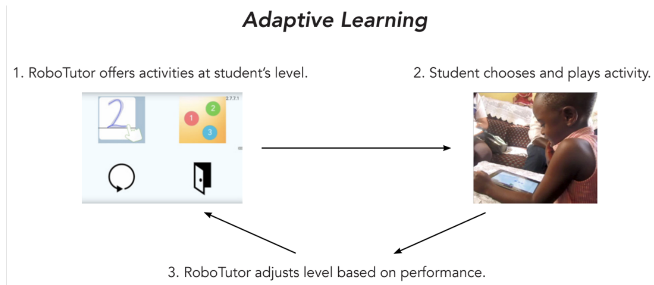


Fig. 2. The process of adaptive learning in RoboTutor

For this project, we chose to analyze how RoboTutor can improve the sequence of activities it provides students. The effectiveness of RoboTutor is measured by metrics that quantify the criteria mentioned above: reliability, usability, engagement, and learning.

This work builds on previously collected RoboTutor log data from two versions of RoboTutor to drive a Multi-Armed Bandit (MAB) algorithm to see which version produces the best student learning results. We consider an arm within the MAB algorithm to be a version of RoboTutor. We measured the learning rate as the number of activities passed per hour according to the percentage of correct student actions. Applying the multi-armed bandit algorithm will allow us to concurrently improve RoboTutor's activity sequence and to analyze the impact of the changes being made.

III. Multi-Armed Bandit (MAB)

MAB was initially developed by Herbert Robins in response to the question, "How should we draw a sample from two populations if our object is to achieve the greatest possible expected value of the sum S=$x_i + ... + x_n$?" [2]

Since then, the multi-armed bandit has developed into a useful tool for addressing complex issues with options. "A bandit problem can be expressed in its most basic form (sometimes referred to as stochastic) as a set of K probability distributions with corresponding expected values and variances. These distributions are typically seen as matching to the arms of a slot machine;" [3] In this typical example of MAB, the gambler eventually wishes to determine which Slot machines offer the best payout.

If there are only two machines, one solution could be to keep pulling the lever on both machines until one machine eventually produces a higher reward over time. However, when hundreds of thousands of slot machines are involved, this approach can become complicated. "At each turn (pull), t = 1, 2,..., the player chooses an arm with index j(t) and receives r(t) Dj as a reward" (t). Bandit algorithms define a strategy for the player to select an arm j(t) at each turn." [3] The MAB algorithm repeatedly selects arms based on a specified exploration vs. exploitation rate. It provides feedback on which arm should be selected after a set number of rounds. This comes down to the "exploration/exploitation" trade-off in machine learning, in which we must simultaneously try new activities to determine which ones are the best and select the best ones so that the student learns. We here adapt such approaches to ITS." [4] The arms for MAB in this study are two different versions of RoboTutor, referred to as CD2 and CD3.

A. Algorithms and Usages

a) Epsilon (e)-greedy: E-greedy. The e-greedy algorithm is widely used thanks to "its simplicity and obvious generalizations for sequential decision problems." Each round, t = 1, 2,... With probability 1-e, the algorithm chooses the arm with the highest empirical mean, and with probability e, it chooses a random arm. To put it another way, given initial empirical means $\mu_1(0), ..., \mu_K(0)$, [3]

$$p_i(t+1) = \begin{cases} 1-\epsilon+\epsilon/k & \text{if } i = \arg\max_{j=1,\ldots,K} \hat{\mu}_j(t) \\ \epsilon/k & \text{otherwise.} \end{cases}$$

Fig. 3. E-greedy mathematical algorithm.

b) Upper Confidence Bound (UCB): "In addition to the empirical means, the simplest algorithm, UCB1, keeps track of the number of times each arm has been played, denoted by ni(t). Each arm is initially played once." [3] In this case, UCB pulls the reward for each version of RoboTutor, as measured by the number of activities passed satisfactorily per hour of use. "Following that, at round t, the algorithm greedily selects the arm j(t) as follows based on:" [3]

$$j(t) = \arg\max_{i=1\ldots k} \left(\hat{\mu}_i + \sqrt{\frac{2\ln t}{n_i}} \right)$$

Fig. 4. UCB mathematical algorithm.

Based on the above function, the UCB algorithm selects the arm (i.e. RoboTutor version). In this case, the argmax function takes the mean up to a specified point. So it's the mean until the next session. The function then takes the log of t and adds the square root of two, which is the upper confidence level, which is the number of times each arm was pulled. This function keeps track of whether RoboTutor version CD2 or CD3 was used. It then divides that total by the number of arms, which is two in this case because we have two versions.

B. Applications of MAB to Intelligent Tutor Systems (ITS)

An intelligent tutor system was tested on "11 different schools in the Bordeaux metropolitan area" in a similar study. We had 400 students between the ages of 7 and 8. This study assessed students' learning by having them take a pre-test a few days before and a post-test a few days after using the ITS. The results of the study show that this [MAB] approach achieves comparable, if not better, learning results than the sequence created by an expert teacher." [1]

IV. Relation to Prior Work With RoboTutor

This project expands on previously collected data and previously written code to sort through the data returned by the RoboTutor tablet. The majority of the code in this project was derived from Yuanhang's Student Analysis scripts and Sanjana's simulation code. Yuanhang's Student Analyses Scripts sorted through all of the log files returned by RoboTutor and compiled them into a csv file based on the previously mentioned four criteria: reliability, usability, engagement, and learning. Yunaghan's csv file contained metrics computed from students' activities in RoboTutor. A few such metrics are in the table below. The complete list can be found in the appendix:

	min	median	max	mean	standard deviation
RELIABILITY					
total crashes	0	0	29	1.35	3.89
crash rate (crash/hr)	0	0	12.15	0.53	1.43
total time to recover(s)	0	0	800	52.13	132.48
ENGAGEMENT					
longevity (# calendar weeks from first to last use)	1	1	182	11.54	29.25
total # days actually used	1	2	76	4.97	8.75
frequency = days used/week used	0.01	1	4	1.29	0.82
USABILITY					
total attempt hiatus(s)	15.69	666.24	61918.76	2310.7	6459.29
min attempt hiatus(s)	0	0	0.57	0.01	0.06
median attempt hiatus(s)	0.01	0.73	2.73	0.81	0.34
LEARNING					
# activities passed per hour of use	0.37	35.47	68.81	34.9	14.33
percent activities passed	20.00%	79.49%	100.00%	77.03%	15.83%
percent PROMOTION activities passed	0.00%	83.92%	100.00%	80.26%	18.67%
percent PLACEMENT activities passed	0.00%	50.00%	100.00%	47.25%	36.68%

Fig. 5. A few measures of reliability, usability, engagement, and learning computed from RoboTutor logs

Below each criterion in the provided snippet of the full spreadsheet are metrics that quantify it. For this project, we used the highlighted cell "number of activities passed per hours of use" to calculate student learning using the UCB multi-armed bandit algorithm.

We computed the minimum, maximum, mean, median, and standard deviation of each metric for each session. The data for each row in the full spreadsheet was recorded for many RoboTutor sessions.

The MAB algorithm is guided by which arm yielded the highest reward. The reward for RoboTutor in this case is learning. The other section of code uses code from Sanjana's simulation experiments. This code ran simulation data through two MAB algorithms (Epsilon Greedy and UCB)

< TABLE OF CONTENTS

and discovered that UCB was the most accurate at predicting which arm had a higher reward. This work will employ the UCB algorithm.

V. Methodology

The MAB algorithm was tested on two RoboTutor versions, CD2 and CD3. The same metrics and computations were used for each version. Coding an MAB approach to compare the two versions of RoboTutor required several steps. MAB requires arms. The arms are the versions in this scenario, so the terms can be used interchangeably.

A. Extracting the version from the log file name

Once the MAB algorithm has been run on the various RoboTutor versions, the output data must specify which version (arm) is being examined. The first section of this project's code extracts the version name from the log files that contain the collected data.

B. Calculating the Arm Rewards

After the version names have been properly extracted, the next step is to create the Arms environment. As previously stated, the arms in this case are versions (CD2 and CD3). The parameters for each arm are stored in an Arm class (mu, sigma, val, and weight). Each arm (version) incorporates the mu and sigma parameters, which in this case are the average and standard deviation discovered when calculating the number of activities passed per hour. Each time an arm is pulled by the UCB algorithm, a method is developed to record the the reward for that arm and compare it to the reward when the other arm is pulled.

A path is created to the RoboTutor data spreadsheets for each version. The code that follows implements a function that sets mu and sigma. When simulating the sum of rewards for each arm, mu was set to the mean number of activities passed per hour for each RoboTutor session, and sigma was set to the standard deviation for number of activities passed per hour for each RoboTutor session. When simulating multi-armed bandit, mu was set to the cumulative mean number of activities passed per hour for each RoboTutor session, and sigma was left as the standard deviation for number of activities passed per hour for each RoboTutor session. The arms are then added to an array to be chosen from.

When simulating multi-armed bandit on the provided data, a method was developed to compute the cumulative average, which represents the average of each RoboTutor session added until the most recent session.

The final section of code is the UCB algorithm, which repeatedly chooses each arm based on the rewards, weight, and uncertainty for each arm. As initial parameters, the UCB algorithm considers the arms and the confidence level. The initial reward for each arm is also zero and t. (number of times an arm has been pulled is set to one). When UCB pulls an arm, it considers the level of uncertainty.

The uncertainty is calculated by adding the upper confidence bound interval to the reward each time an arm is called. The final step is a method called Makedecision, which is used to pull one of the two arms. The reward value is updated every time an arm is pulled.

VI. Results

For the first part of the results, we simulated a total of rewards for each arm using the data collected, yielding a total of activities passed per hour of use.

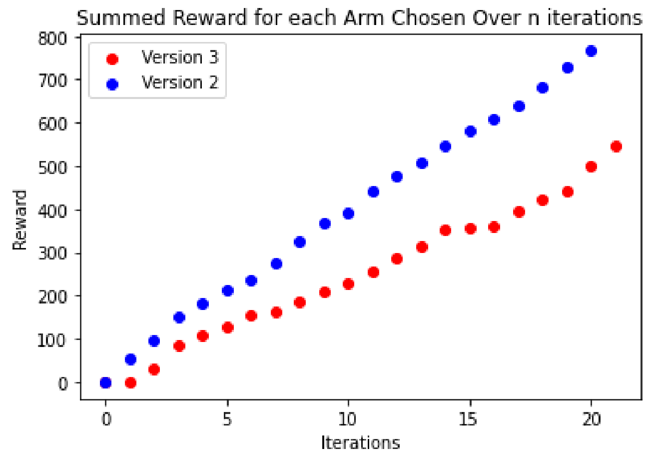


Fig. 6. Average cumulative reward value over n iterations of RoboTutor sessions

Over time, the reward value, which is the number of activities passed per hour of use, significantly outperformed CD2, indicating that CD3 had a higher reward value. The graphs below depict simulations of multi-armed bandit on the given data using the UCB algorithm. The UCB algorithm for multi-armed bandits pulls each arm over time based on some exploration vs exploitation and eventually pulls the arm with the highest cumulative reward value (number of activities passed per hour of use).

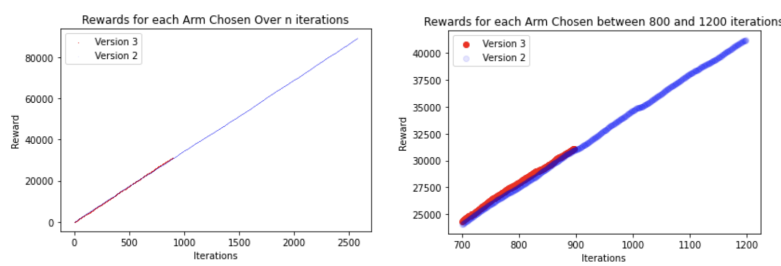


Fig. 7. Simulation of Multi-armed bandit

The graph on the left shows that UCB was pulling both versions of RoboTutor up until around 800 iterations. Then it discovered from itself that CD3 yielded a higher reward. As a result, the red line is no longer used. The graph to the right is a magnified version of the graph to the left. The right graph has been zoomed in to show where UCB learns to start selecting CD3.

Because these plots show each reward at the time when each arm is pulled, the results of the simulated sum of rewards and the simulated MAB behavior indicate that CD3 of RoboTutor performed better and had a higher overall average amount of learning for each session collected.

VII. Discussion

A challenge in this project was distinguishing between the historical and simulated rewards. The historical reward is the average that was given based on the column values from the analysis spreadsheet, whereas the simulated reward is what we actually computed, and it represents how the reward would grow over time. We had to consider how we are constrained by the data we currently have. This multi-armed bandit strategy is only applied to data that has been provided to us. It does not account for new data, which can be a problem because new data may change which arm should be chosen. In the future, we'd like to find a way to run multi-armed bandits on RoboTutor at run time so that it can account for any incoming data. We also looked at how many iterations of running the UCB algorithm it took for the two versions of RoboTutor to converge. 7 demonstrates that it took approximately 800 iterations for the UCB algorithm to determine that CD3 performed better than CD2.

VIII. Future Work

Calculating how much data would determine which arm is best by calculating the effect size on each arm would be a step toward a more accurate analysis of RoboTutor. The effect size would provide a definite number indicating which arm is superior. Developing a more accurate function for measuring a student's learning with RoboTutor would provide a more valuable understanding of how well students learn with RoboTutor. The current learning measurement is the number of activities passed per hour. This metric does not provide objective evidence that a student is learning. The metric "percentage of promotion activities passed" would be used for objective learning. This metric indicates the number of activities performed in PROMOTION mode rather than PLACEMENT mode. Each time a student completes a more complex activity than the previous one, the percentage of promotion activities passed is calculated. Given that if a student completes PROMOTION activity, they have learned from the previous activity, the percentage of promotion activities passed could be a more accurate metric for learning.

APPENDIX

Link to Yuanhang Wang's Student Analysis Code - `https://colab.research.google.com/drive/1Xd6KZbAxb8MU6WvfGsUqIq6nGuQrlzjn?authuser=1`

RoboTutor Data Spreadsheets must be downloaded and uploaded as files into the below google colab notebooks to run properly -

CD2 dataset - `https://docs.google.com/spreadsheets/d/1CIPL-JJz_IoYYwgdw24e22AR_yykf2s94QA7LfOFLe4/edit#gid=399644296`

CD3 dataset - `https://docs.google.com/spreadsheets/d/1DhfpkclpkOFRUP3OORB3ZDK5v_12uCtTBmw3H3NWir8/edit#gid=651261727`

Code for multi-armed bandit for simulated rewards - `https://colab.research.google.com/drive/1roLh5aUni3oHirD4jcbRMnCPlGrZRDcF?authuser=1#scrollTo=24compleoOTz0q1jIt`

Code for simulating multi-armed bandit on given RoboTutor data - `https://colab.research.google.com/drive/15bPUDegz-mKngpRdXAq3yad9t6zj_i0y?authuser=1#scrollTo=24oOTz0q1jIt`

ACKNOWLEDGMENT

This work would not have been possible without the personal and professional mentor-ship of Carnegie Mellon University Emeritus Research Professor Jack Mostow. This material was also funded and supported by Director John Dolan, and Co-Director Rachel Burcin of The Robotics Institute of Summer Scholars within The Robotics Institute at Carnegie Mellon University. I would also like to thank the Robotics Institute for funding this enriching learning opportunity.

References

[1] B. Clement, D. Roy, P.-Y. Oudeyer, and M. Lopes, "Multi-armed bandits for intelligent tutoring systems," *arXiv preprint arXiv:1310.3174*, 2013.

[2] H. Robbins, "Some aspects of the sequential design of experiments," *Bulletin of the American Mathematical Society*, vol. 58, no. 5, pp. 527–535, 1952.

[3] V. Kuleshov and D. Precup, "Algorithms for multi-armed bandit problems," *arXiv preprint arXiv:1402.6028*, 2014.

[4] B. Clement, D. Roy, P.-Y. Oudeyer, and M. Lopes, "Online optimization of teaching sequences with multi-armed bandits," in *7th international conference on educational data mining*, 2014.

AirLoc: Object-based Indoor Relocalization

Aryan[1], Chen Wang[2], and Sebastian Scherer[2]

***Abstract*— Indoor relocalization is vital for robotic tasks such as autonomous exploration and semantic scene understanding. Many previous approaches attempt geometry-based methods to carry out indoor relocalization, but these methods are not robust. Objects are more informative than geometry elements, and places are often unique due to the placement of objects. The critical challenges in object-based relocalization are object reidentification and remembering object relationships. In this context, we propose a novel object-based indoor relocalization approach, dubbed AirLoc. We use objects' appearance and geometric relationships to extract qualitative information which is utilized to perform higher-level task of relocalization. Specifically, we use object embeddings to get appearance based room similarity and pixel location based features to get geometry based room similarity which are collectively used for room level localization. We demonstrate that AirLoc performs very well for room reidentification and is robust to severe occlusion, perceptual aliasing, viewpoint shift, deformation, and scale transformation. To the best of our knowledge, AirLoc is one of the first object-based indoor relocalization approach.**

***Index Terms*— Indoor Re-Localization**

I. INTRODUCTION

Recently, indoor relocalization has gained unprecedented attention with the development of numerous location-based mobile phone and robotic applications such as augmented reality (AR) [1], mobile robot navigation, and simultaneous localization and mapping (SLAM) [2], [3].It can be employed in large indoor buildings such as shopping malls, offices where one can use his cell phone to relocalize himself when lost. Several existing mobile robot localization techniques, such as visual odometry and simultaneous localization and mapping (SLAM), require indoor relcalization to correct their drift. The robot continues its work even if its tracking fails by performing global relocalization to recover the camera's pose estimation. This problem is sometimes known as the kidnapped robot problem also, where the task is to localize a robot in an unknown environment when a prior estimate of the location is unavailable.

In recent years, methods have been developed to address the indoor relocalization problem [4]. Some are keyframe based with store keyframe images and some are feature based which store keypoints extracted from keyframe images. However, these algorithms either require a 3D model of the scene or multiple database images to get good results. But getting such 3D models or multiple database images is not easy in most real-world indoor applications, such as large offices and supermarkets. Apart from it, several methods do not work well in situations, such as occlusion, light changes, and the interference of personnel access. It remains questionable whether illumination changes and occlusion can be compensated while maintaining only limited overlap between the localization query and database. Therefore, rather than merely concentrating on geometric mapping, semantic mapping also strongly needs to be considered.

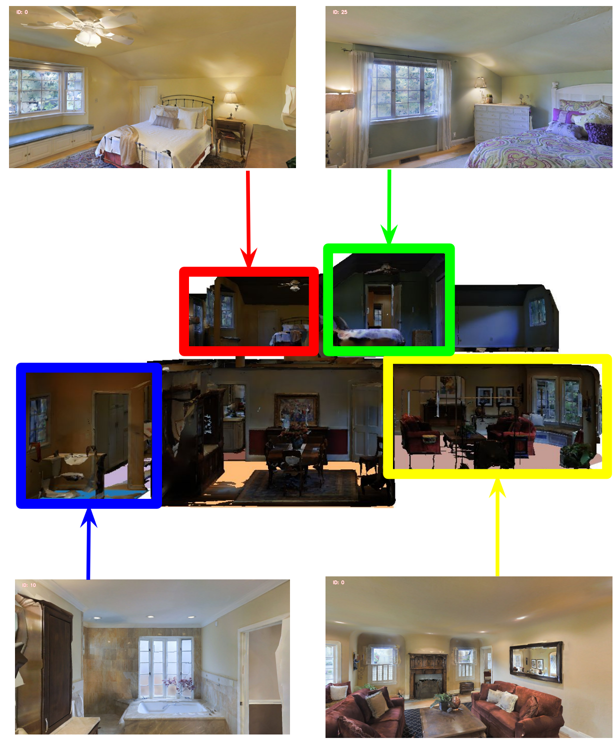


Fig. 1. Indoor Relocalization

Researchers have shown increased performance for object encoding and re-identification tasks in recent years. As objects are more informative than geometry elements, and the location of objects makes spaces often distinctive. We propose AirLoc: an object-based indoor relocalization approach where we show that object similarity and relative object geometry can be a highly efficient technique to relocalize a query given a database of rooms. Further, a room embedding can be obtained from a minimal number of database images and even from a single image, reducing the database size required for relocalization.

In summary, the main contributions of this paper are:

- We introduce AirLoc, a simple yet effective indoor relo-

[1]Aryan is with the Department of Electronics An Communication, Delhi Technological University, Delhi, India aryan_2k19ec032@dtu.ac.in

[2]Chen Wang and Sebastian Scherer are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA chenwang@dr.com; basti@andrew.cmu.edu

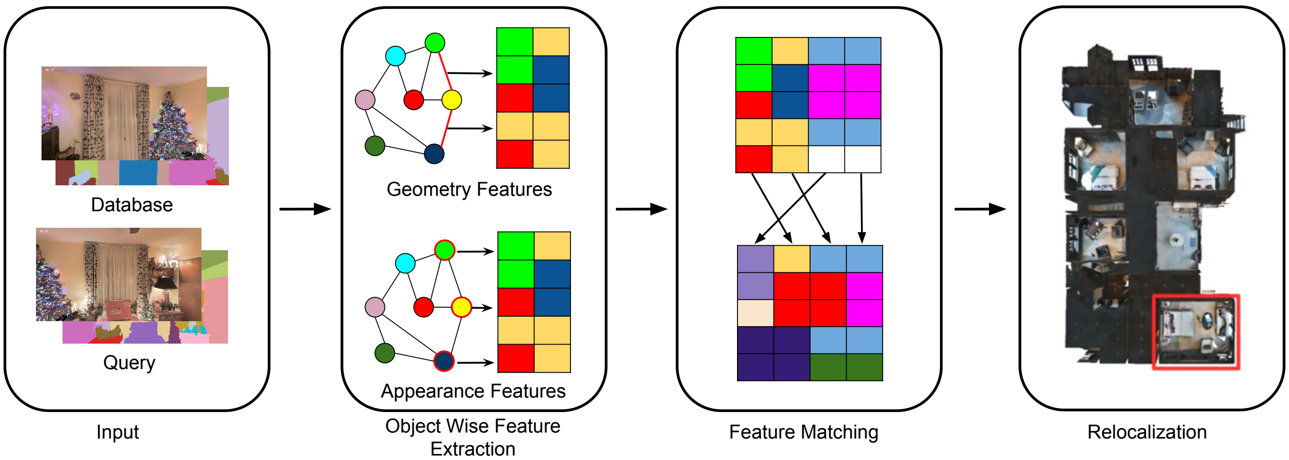


Fig. 2. The proposed object matching framework, which matches query objects with database objects to generate a room similarity and uses the room similarity for room level localization

calization approach that utilizes geometric and semantic information using objects in a scene.

- We extract appearance based features using object encoders and geometry based features using the novel geometry module which extracts relative position between objects. Finally, we perform relocalization using the matching module which performs object and room level descriptor matching.
- AirLoc gives reliable results and outperforms the state-of-the-art methods in room-level localization task. We evaluate AirLoc on custom generated Mp3d dataset which contains room level divisions, depth and semantic data for every image.

II. Related Work

This section reviews 6-DOF pose estimation frameworks and feature-based representation methods. Furthermore, we also review localization methods based on object-level data association.

A. 6-DOF Pose Estimation

Recent years have seen an increased interest in convolutional neural networks (CNN)-based visual positioning algorithms. Using neural networks, PoseNet [5] provides end-to-end visual positioning. It directly performs 6-DoF camera pose regression on input RGB images using a fully connected layer based on GoogleLeNet [6]. In large-scale outdoor scenes, it can achieve 2m and 3°accuracy, while in indoor scenes, it can achieve 0.5m and 5°accuracy. In subsequent years, many improved versions have been developed based on this idea. Walch and Hazirbas [7] add LSTM after the fully connected layer to solve the problem of over-fitting, which could improve the positioning accuracy by 32–37%.

To perform pose regression and odometry on continuous images, Valada and Radwan [8] proposed a multi-task model of a CNN. Hard parameters were shared between the global pose and the odometer subnetwork. A consistent pose estimation can be achieved by limiting the search space by the relative pose between the two images. This method improved positioning accuracy significantly and approached close to traditional local feature-based positioning. Based on [8], [9] added semantic information to the localization stream by aggregating motion-specific temporal information and adding semantic information together with 6-DoF pose regression and odometer information.

Geometric consistency requirements have also been utilized to increase the precision of pose regression, and they have been found to be more effective than employing only euclidean distance constraints. Using depth image data, [10] simultaneously adds the geometric similarity error and photometric error to the loss function. The reprojection error was first presented by [11], who projected the 3D point in the environment onto the 2D picture plane using the camera's predicted pose and used the pixel location deviation as a constraint.

These methods require a lot of training data before inference and require model retraining for every new environment. In this context, our proposed object-based framework relies on the object encoders, which are robust to environmental changes. Hence, the framework can be easily applied to new environments without retraining.

B. Feature-based representation methods

In conventional methods such as loop closure detection, object matching, and VPR, handcrafted features such as SIFT [12] and SURF [13] have been widely applied. Fast appearance-based mapping (FABMAP) [14] identifies revisited objects through feature distribution matching in the trained visual vocabulary of SURF features. A binary descriptor ORB [15] was utilized in DBoW2 [16] to extend this concept further and increase speed. Several approaches [17]–[19] have been suggested to expand on the idea of vocabulary-based retrieval. However, when the local descriptors are not discriminative, these handcrafted feature-based algorithms become susceptible to environmental changes and produce false matches.

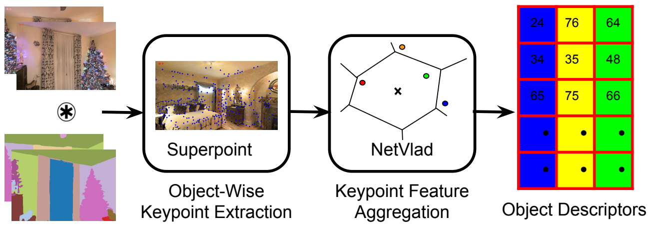


Fig. 3. Appearance based database creation pipeline: Database images along with their semantic labels are passed through a object descriptor to generate object descriptors for objects present in the room

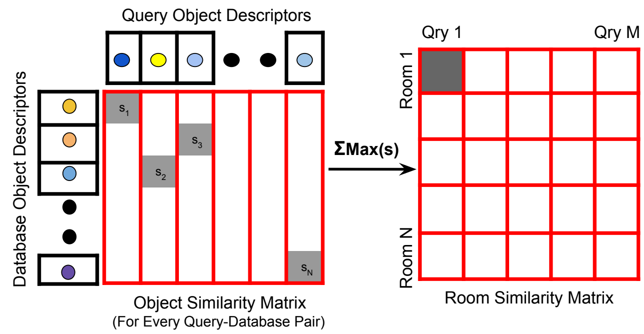


Fig. 4. Appearance-Based Matching: Maximum Object Similairty for every query-databse pair is summed up to form room similarity. Room similarity is further used for localization

Compared to handcrafted features, approaches using deep learned features have demonstrated enormous improvements. One such technique, [20], generates CNN features that are viewpoint invariant using a multi-scale feature encoding across two CNN architectures, significantly improving performance. NetVLAD [21] is a similar method, which uses a Generalized end-to-end deep learning-based VLAD layer, inspired by the "Vector of Locally Aggregated Descriptors". Incorporating spatial/depth data into the RGB domain for object recognition has been addressed using several techniques [22]–[24], which have further explored various input modalities such as RGB-D pictures and point cloud data.

SuperPoint [25], a recently proposed deep learning method, uses self-supervised learning for training interest point detectors and descriptors. Expanding upon SuperPoint, SuperGlue [26] introduced a graph neural network that matches two sets of local features by jointly finding correspondences and rejecting non-matchable points. For the tasks such as feature matching and hierarchical VPR, both SuperPoint and SuperGlue have received widespread adoption [27], [28].

C. Object Based Methods

In order to bridge the gap between perception and action, the robotics community has taken a keen interest in semantic SLAM. The pioneering work of SLAM++ [29] performs object-level SLAM using a depth camera. The main restriction is that an object database of 3D shape and global description must be built in advance. [30] develop a quadratic-programming-based semantic object initialization scheme to achieve high-accuracy object-level data association and real-time semantic mapping. [31] integrated object detection and localization module together to obtain the semantic maps of the environment and improve localization. X-View [32] can localize aerial-to-ground globally and ground-to-ground robot data of drastically different viewpoints using graph descriptors based on random walks.

Recently, AirCode [33] proposed a feature sparse encoding and object dense encoding method which is robust to viewpoint changes, scaling, occlusion, and even object deformation. Building upon that, AirObject [34] uses a temporal CNN across structural information of multiple frames obtained from a graph attention-based encoding to perform temporal 3D object encoding. However, using these object descriptors for relocalization still remains an open question. Taking motivation from above examples, we use object encoders such as AirCode to extract object embeddings which are further used in relocalization.

III. Proposed Approach

We propose AirLoc, a new architecture shown in Fig. 2, which uses qualitative object features for indoor relocalization . In this section, we will first explain the object-encoding frameworks with a comparison among all the available object encoders. Then, we will present the appearance and geometry modules and their ensembling. Finally, we will discuss the loss function used for training the geometry module.

A. Object Encoders

Intuitively, keypoint features and their descriptors on an object are distinctive features of the object. Considering previous works from the literature, we believe these keypoint features can provide robust object embeddings. Hence based on this hypothesis, we use keypoint features and try different aggregators to aggregate the keypoint features into a single embedding vector. The object encoder is show in Fig. 3

We extract feature points on the objects using point detector SuperPoint [25], where the position of each feature point is denoted as $p_i = (x, y), i \in [1, N]$, and the associated descriptor as $d_i \in R^{D_p}$, (D_p is the descriptor dimension). To segment objects in an image, we either use ground-truth instance segmentations or masks from commonly used networks like Mask R-CNN [35] or an open-world object detector [36]. Given these object-wise grouped feature points, we aim to aggregate the individual point features to form a collective object encoding. We try different models for this aggregation. Firstly, we try a graph attention encoder by using the keypoint descriptors as nodes and thus forming a topological graph representation. We pass this graph through two layers of the graph attention network and use the arithmetic mean to aggregate all the nodes into an object embedding. Similar to what we have done in the case of Graph Attention, we try a different model, i.e., graph convolution network as well.

Graphs perform well if the training data and test data are from the same environment. However, if the train and

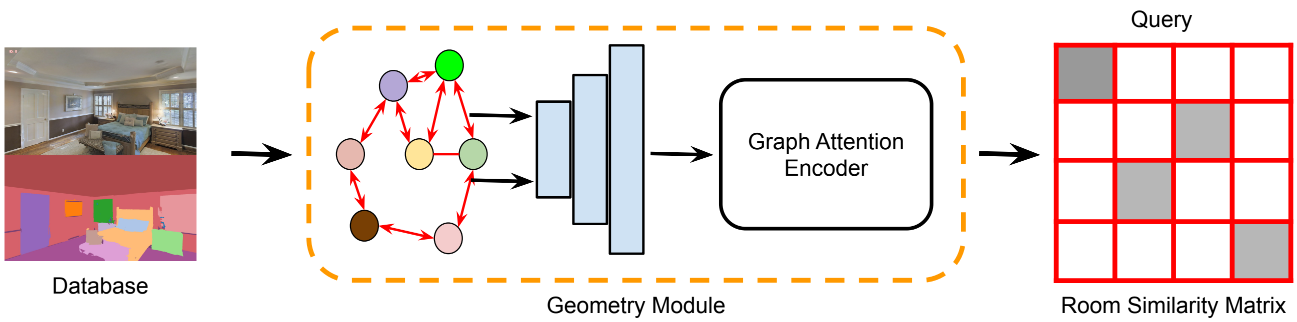


Fig. 5. Geometry Based Matching: Geometry Module is used to extract room wise feature for both database and query with is further used to create Room Similairty Matrix

test data are not from the same environment, the graph models tend to overfit. To solve this problem of overfitting, we use NetVLAD, one of the most commonly used image retrieval frameworks. The NetVLAD model we use for our experiments on Mp3D dataset is trained on COCO [37] and YT-VIS [38] dataset but it performs well on Mp3d as well, indicating its robustness to environmental changes.. Given N Superpoint descriptors of dimention D_p and K cluster centres output of NetVLAD V is a $K \times D_p$ dimentional vector given by:

$$V(k) = \sum_{i=1}^{N} a_k(x_i)(x_i - c_k) \tag{1}$$

where x_i and c_j are i-th Superppoint descriptor and k-th Cluster centre respectively. $a_k(x_i)$ is the learnable parameter that denotes the soft assignment of descriptor x_i to cluster c_k. The matrix V is first intra-normalized and then converted into a vector. The vector of is again L2-normalized to form the output vector O .

B. Appearance Module

1) Database Construction: We show results on room-level localization, where given a query image, the task is to localize it to the room to which it belongs. In order to scale the model to new environments, the framework should not need a large number of images from the new environment as a database. In this context, we just take a few (K) images (1,2,5,10) from every room to create our database. The images are further passed to an object encoder described above to generate object embeddings for the objects present. If the same object is present in multiple images, we take the arithmetic mean over multiple object embedding. Finally, we get a list of object embeddings for all the objects in the room. The database creation framework is shown in Fig. 3.

2) Relocalization: Finally, after generating the database, we use the novel matching architecture to match the query image with the database. Given a query, we first extract all the query object descriptors using the same method we used for database collection. Once we have the desired object descriptors for the database and query, we perform object matching, using cosine similarity as matching metric. After exhaustively matching the query objects with the database objects, we get an object similarity matrix S where each column contains a matching score of a query object with all the database objects. This can be conputed as :

$$S(m, n) = O_d(m) \cdot O_q(n) \tag{2}$$

where m and n are m-th database object and n-th query object. O_d is database object embeddings and O_q are query object embeddings and $\cdot$ is used to denote dot product.

The database object having a maximum similarity score with a query object is considered a match. We sum up the cosine similarities of the matched objects and form a query-database room similarity score.

$$R(p, q) = \sum_{n=0}^{N} \max(S_{pq}(m, n)) \tag{3}$$

where m and n are m-th database object and n-th query object. p and q are p-th database room and q-th query room. S_{pq} is the the object similairty matirx for p-th database room and q-th query room. R is the room similairty matrix. Finally, after computuing room similairty matrix, the room having maximum similarity with the query is considered a match. The matching framework is shown in Fig. 4.

C. Geometry Module

It is observed that sometimes two rooms have similar objects but are placed in different relative positions. The above proposed appearance based method may not be able to correctly classify rooms in such situation. Hence, there is a need to incorporate some geometry based information to assist appearance based matching in such kind of situations. We propose a novel geometry-based learning approach where we learn the relative positive between two objects and use it for relocalization. This approach is shown in Fig. 5

Firstly, we extract the location features of every object which are mean pixel location (μ_i), standard deviation pixel location (σ_i) , 1st, 2nd and 3rd Order Moment of pixel location (m_i^1, m_i^2, m_i^3) , singular Value decomposition of pixel location (svd_i). These locations are passed through a Multi Layered Perceptron (MLP) to get absolute location features of dimention E_i. The final object-wise absolute location features are subtracted from each other to get relative object features. In this way, if there are m objects we get ${}_mC_2$ relative location features between them. These features can be computed as:

$$v_i = [\mu_i, \sigma_i, m_i^1, m_i^2, m_i^3, svd_i] \tag{4}$$

$$e_{ij} = g(v_i) - g(v_j) \tag{5}$$

where [] is concatenation, e_{ij} is the relative location feature between i-th and j-th object and g() denotes MLP layer.

These relative location features are then passed through a two-layered Graph Attention Encoder to enable structured attention-based message propagation between the location features. This evolves a global sense into the location features using global interaction. Afterwards, the updated node features are average pooled to get graph features corresponding to a room. These room level embeddings can be obtained as:

$$e_i^t = \sigma \sum_{j \in \mathcal{N}(i)} a_j \cdot W \cdot e_j^{t-1} \tag{6}$$

$$r = \frac{\sum_{i=0}^{N} e_i}{N} \tag{7}$$

where e_i^t is the i-th location feature at t-th graph layer. a_j is the learnable attention weight [39]. r is room level embedding of dimention E_o.

Once we have a room level embedding of database and query, we use cosine similarity to get a room similarity matrix analogous to appearance based matching technique. Every element of the room similarity matrix is cosine similairty of the corresponding query and database room.

$$R_{loc}(p, q) = r_p \cdot r_q \tag{8}$$

where r_p and r_q and p-th database and q-th query room.

D. Ensembling Appearance and Geometry

Now, as we have two room similarity matrices (Appearance based and geometry based), we can ensemble them for final matching. We use weighted sum technique where we add the two room similarity matrices with some weights to get the final room similarity matrix (R') used for classification.

$$R' = m \cdot R + R_{loc} \tag{9}$$

Furthermore, it is observed that the most of the appearance based false matches have little difference in highest and second highest room similarity. True matches from the appearance based methods generally have a large gap between highest and second highest similarity. Hence, we apply geometry based assistance only to those queries where the difference between the similarity of highest and second highest match is less than some threshold which we call as appearance threshold(T_{diff}). The classifications having this difference greater than threshold are classified using appearance matching only.

E. Loss Function

The graph attention encoder in geometry module is supervised by the room matching loss. The room matching loss L_r maximizes the cosine similarity of positive room pairs and minimises the cosine similarity of negative room pairs.

$$\begin{aligned} L_m = & \sum_{\{p,q\} \in P^+} (1 - C(R_p, R_q)) \\ & + \sum_{\{p,q\} \in P^-} \max(0, C(R_p, R_q) - \zeta) \end{aligned} \tag{10}$$

where $\zeta = 0.2$ is a constant margin, S is the cosine similarity, and P^+, P^- are positive and negative object pairs, respectively.

IV. Experimental Results

A. Dataset

1) Background: Many datasets have been collected for semantic understanding of scenes. The Places365-Standard dataset contains 1.8 million train images from 365 scene categories. There are 50 images per category in the validation set and 900 images per category in the testing set. ADE20k dataset contains 20,210 images in the training set, 2,000 images in the validation set, and 3,000 images in the testing set. All the images are exhaustively annotated with objects. Many objects are also annotated with their parts. For each object, there is additional information about whether it is occluded or cropped. MIT Indoor scenes database contains 67 Indoor categories and a total of 15620 images. The number of images varies across categories, but there are at least 100 images per category. A recently introduced indoor RGB-D dataset of changing indoor environments is RIO10. It consists of 74 sequences. They provide splits into training, validation (one sequence per scene), and testing sets, leaving us with ten train, ten validation, and 54 test sequences overall.

Considering the task of object-based scene understanding and using the encoded scene for re-localization, the dataset must contain hierarchical scene labels such as room level and building level labels for the images. However, none of those mentioned above datasets contain such hierarchical labels. Apart from that, there are some particulars essential for better learning such as :

- Ground truth semantic segmentation and depth labels
- Multiple images from the same scene but a varying viewpoint
- Ground truth 6D pose data for every image.

Existing datasets miss at least one of the above-described characteristics. So there is a need for a dataset with Hierarchical labels and all the specifications mentioned above.

B. Dataset Generation

We require multiple images from the same scene but varying viewpoints, and the best way to generate such a dataset with ground truth labels is to use a 3D simulator. We use Habitat-Sim, a high-performance physics-enabled 3D simulator supporting 3D scans of indoor/outdoor spaces

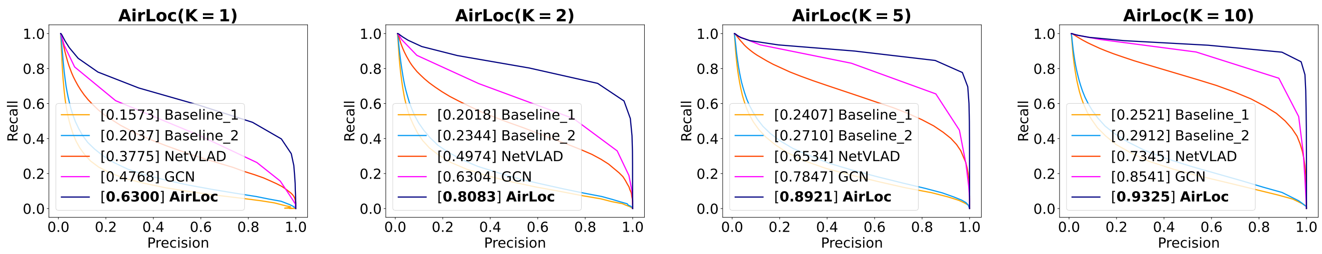


Fig. 6. Precision-Recall plots showing comparison between AirLoc and different baselines for different K values

and rigid-body mechanics. We want the simulation to real-world gap as less as possible, and for this purpose, we use Matterport3D, a large-scale RGB-D dataset that contains 90 building-scale scenes. All the scenes are in the form of textured 3D meshes and are created from 194,400 real-world RGB-D images. Annotations are provided with surface reconstructions, camera poses, and 2D and 3D semantic segmentations.

We select 15 scenes from the dataset, and for every room present in a scene, we extract approx 2500 random navigable poses. Here navigable pose means a pose that is easily navigable for a human or a robot, i.e., not present inside a wall or below the ground. Hence, by collecting these poses, we are making sure that the images which will further be generated from these poses are similar to what humans or robots perceive in thier general actions. We then generate corresponding RGB image, semantic segmentation and depth images for all the collected poses. In this way, the dataset is divided into buildings and rooms. We further divide the dataset into test and train split as well where 80% images from every room are train split and remaining are test split.

C. Implementation Details

The AirLoc configurations for appearance based matching are superpoint input dimention D_p = 256 and NetVLAD number of clusters K = 32. To test the generalizability of method, NetVLAD is not trained on the Mp3D dataset. Configuration for geometric matching are reltive position feature dimention E_i = 256, hidden dimention of graph layer E_h = 512, output dimention of grpah E_h = 1024. For GAT we use 8 number of heads and dropout of 0.5. For training, we employ a batchsize of 256, learning rate of $1e^{-4}$ and Adam optimizer.

D. Evaluation Criterion

For evaluation of room level localization performance, we use the test split of Mp3d dataset. To switch between appearance only and appearance-geometry matching, we use the difference in highest and second highest room similarity(T_{diff}) as 0.1. For appearance-geometry matching the weight (m) for the weighted sum of appearance and geometry is 10. We calculate the accuracy as the ratio of correctly classified rooms to total number of rooms.

E. Comparison to State-of-the-art Methods

We compare our proposed approach with two kinds of benchmark baselines. First kind of baselines extract room level features directly from query and database and then match them to generate room matching scores. Second kind of baselines extract object information from query and database and match object level data to generate room similarity scores. The room level baselines are Baseline_1, Baseline_2, NetVLAD. The object level baseline is gcn.

TABLE I
THE RESULTS COMPARING AIRLOC WITH BASELINES

Method	Accuracy				
	K=1	K=2	K=3	K=5	K = 10
Baseline_1	31	50	59.27	65.54	75.24
Baseline_2	34	41.9	48.43	51.84	57.58
NetVLAD	50.9	65.3	75.65	84.22	91.45
GCN	56	71.5	80.66	86.84	93.33
AirLoc	**65**	**82.2**	**89.17**	**93.16**	**97.17**

For Baseline_1 we first extract NetVLAD based object descriptors for the input images and then a room level embedding is obtained by averaging over the object descriptors. Baseline_2 is similar to Baseline_1 except for the fact that we use a Graph Attention Network as object encoder instead of NetVLAD. For NetVLAD baseline, we pass the input images through a NetVLAD layer and use the output descriptors as room level features. In gcn baseline, we use the similar framework as AirLoc but replace NetVLAD with a two layers Graph Attention Network.

Table I and Fig. 6 contain qualitative comparisons of AirLoc and baseline methods. In terms of accuracy and PR-AUC, AirLoc exceeds all other approaches hands down. It can be seen that AirLoc performs well, especially outperforming the best baseline by an average of 12% in PR-AUC% and 8% in accuracy. Additionally, for both the metrics, the performance gap between object-based approaches and room-based methods is consistently large, demonstrating the importance of object level data.

V. ACKNOWLEDGEMENT

This work was supported by OPPO and the AirLab at Robotics Institute, Carnegie Mellon University as a part of Robotics Institute Summer Scholars (RISS) program. Special thanks to Ms. Rachel Burcin and Dr. John M. Dolan for their support throughout the program.

< TABLE OF CONTENTS

REFERENCES

[1] T. Khan, K. Johnston, and J. Ophoff, "The impact of an augmented reality application on learning motivation of students," *Advances in Human-Computer Interaction*, vol. 2019, 2019.

[2] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[3] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16 558–16 569, 2021.

[4] A. Majdik, M. Popa, L. Tamas, I. Szoke, and G. Lazea, "New approach in solving the kidnapped robot problem," in *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*. VDE, 2010, pp. 1–6.

[5] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[7] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using lstms for structured feature correlation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 627–637.

[8] A. Valada, N. Radwan, and W. Burgard, "Deep auxiliary learning for visual localization and odometry," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 6939–6946.

[9] N. Radwan, A. Valada, and W. Burgard, "Vlocnet++: Deep multitask learning for semantic visual localization and odometry," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4407–4414, 2018.

[10] M. Tian, Q. Nie, and H. Shen, "3d scene geometry-aware constraint for camera localization with deep learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4211–4217.

[11] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5974–5983.

[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[13] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[14] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[15] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[16] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[17] E. Garcia-Fidalgo and A. Ortiz, "ibow-lcd: An appearance-based loop-closure detection approach using incremental bags of binary words," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3051–3057, 2018.

[18] M. Gehrig, E. Stumm, T. Hinzmann, and R. Siegwart, "Visual place recognition with probabilistic voting," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3192–3199.

[19] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2548–2555.

[20] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford, "Deep learning features at scale for visual place recognition," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3223–3230.

[21] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[22] M. Schwarz, H. Schulz, and S. Behnke, "Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 1329–1335.

[23] H. F. Zaki, F. Shafait, and A. Mian, "Convolutional hypercube pyramid for accurate rgb-d object category and instance recognition," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1685–1692.

[24] ——, "Viewpoint invariant semantic object and scene categorization with rgb-d sensors," *Autonomous Robots*, vol. 43, no. 4, pp. 1005–1022, 2019.

[25] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[26] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[27] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 716–12 725.

[28] N. V. Keetha, M. Milford, and S. Garg, "A hierarchical dual model of environment-and place-specific utility for visual place recognition," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6969–6976, 2021.

[29] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1352–1359.

[30] Z. Qian, K. Patath, J. Fu, and J. Xiao, "Semantic slam with autonomous object-level data association," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 203–11 209.

[31] L. Zhang, L. Wei, P. Shen, W. Wei, G. Zhu, and J. Song, "Semantic slam based on object detection and improved octomap," *IEEE Access*, vol. 6, pp. 75 545–75 559, 2018.

[32] A. Gawel, C. Del Don, R. Siegwart, J. Nieto, and C. Cadena, "X-view: Graph-based semantic multi-view localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1687–1694, 2018.

[33] K. Xu, C. Wang, C. Chen, W. Wu, and S. Scherer, "Aircode: A robust object encoding method," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1816–1823, 2022.

[34] N. V. Keetha, C. Wang, Y. Qiu, K. Xu, and S. Scherer, "Airobject: A temporally evolving graph embedding for object identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8407–8416.

[35] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[36] K. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian, "Towards open world object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5830–5840.

[37] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[38] L. Yang, Y. Fan, and N. Xu, "Video instance segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5188–5197.

[39] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

Uncertainty Quantification for Image Segmentation

Brad Shook[1], Ifigeneia Apostolopoulou[2], Jieshi Chen[2], and Artur Dubrawski[2]

Abstract— **During the last decade, deep learning has yielded unprecedented results in computer vision. However, the decisions made by these models have inherent uncertainties associated with them. In safety-critical applications where certainty is paramount, like autonomous driving, these uncertainties are vital. Due to the importance of uncertainty quantification in machine learning models and the lack of code and data availability, Google Research developed a standardized framework containing state-of-the-art methods, datasets, and evaluation metrics. In this paper, we convert the image segmentation architectures and methods from Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision to Google Research's framework and reproduce their results. The conversion is ongoing so we present preliminary results for a single model at this time.**

***Index Terms*— Deep Learning Methods, Computer Vision for Automation, Segmentation**

I. INTRODUCTION

The uncertainties associated with deep learning models are of utmost importance in safety-critical applications such as health care and autonomous driving where models must be certain about their decisions [1] [2]. This has led to increased research into methods for quantifying uncertainties inherent to deep learning models and their data [3]. However, there is a lack of code and data availability in many uncertainty quantification (UQ) studies which leads to issues with reproducibility [4].

This issue led Google Research to create a framework [5] that implements state-of-the-art UQ models, datasets, and evaluation metrics. The framework allows researchers to extend current methods while providing baselines to compare with. One important task that is not included in the framework is image segmentation, which involves identifying what class each pixel in an image belongs to. For example, when given an image from a car dash camera, the model aims to label each pixel as belonging to a car, pedestrian, road, etc. Given the importance of UQ for this task in applications such as medical imaging [6] and crash-prevention in cars [2], it is crucial that the repository is expanded to include this task in order to provide a baseline for researchers to build off of.

Prior works focusing on UQ for image segmentation have primarily implemented Monte-Carlo (MC) dropout [7] and ensembling [8]. Additionally, Bayesian neural networks [9] are able to quantify uncertainties. These methods have advantages and disadvantages so it is important that comparisons can be made between them.

In this paper, we expand upon the work of [5] by converting the PyTorch [10] training and evaluation pipeline for image segmentation from [11] into a TensorFlow [12] version that works with [5]. Then, we demonstrate preliminary results of our implementation and discuss the current issues with our implementation.

II. RELATED WORKS

A. Ensembling and MC-Dropout

Ensembling involves training k models where each model is trained on bootstrapped samples of the original training set. By aggregating the outputs of the models in an ensemble, we are able to quantify the predictive uncertainty. The parallel nature of ensembling helps with time complexity, but increases the computational resources needed for training. This method has proven to be a strong approach for predictive uncertainty estimation [8].

While ensembling reduces time complexity, the benefit of MC-dropout is that it trains a single model. In MC-dropout, dropout layers are used in the model architecture [4]. In these dropout layers, a percent of random nodes are dropped from the network. Dropout is often used to combat overfitting in neural networks but also allows the single model to act as many different models since different sets of neurons are dropped each forward pass through the network. In [7], it is demonstrated that by gathering outputs from k forward passes through a network, the predictive uncertainty of the model can be captured.

B. Evaluating Scalable Deep Learning Methods

In [11], scalable deep learning methods for uncertainty quantification in computer vision tasks are investigated. There are two types of uncertainty, epistemic and aleatoric. Epistemic uncertainty measures the inherent uncertainty in the model. This type of uncertainty can lead to over-or-under confidence in the model's predictions. Aleatoric uncertainty refers to the uncertainty in the training data. Both ensembling and MC-dropout are able to quantify these uncertainties [11].

These UQ methods were tested on the Cityscapes dataset and a synthetic dataset. For each uncertainty metric, ensembling consistently outperformed MC-dropout. The authors identified the simplicity of ensembling as another added benefit. However, they noted the main drawback present in both methods is the computational cost at test time, limiting the applicability of these methods in time-critical tasks.

[1]Brad Shook is with the Computer Science Department, Davidson College, Davidson, NC, USA brshook@davidson.edu

[2]Ifigeneia Apostolopoulou, Jieshi Chen, and Artur Dubrawski are with the Auton Lab, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA iapostol@andrew.cmu.edu, jieshic@andrew.cmu.edu, awd@andrew.cmu.edu

III. METHODOLOGY

A. Training Data

The Cityscapes [13] dataset is often used for semantic segmentation of urban scenes. Images contained in the dataset were captured from 50 different cities and were not captured in adverse weather [13]. Each image is 1024 by 2048 and has a corresponding pixel-level annotation. Annotations include 30 classes, ranging from human to void. However, only 19 of these classes are used in training and evaluation, in order to match with [11]. The other 11 classes are ignored. The dataset includes 2975 train images, 500 validation images, and 1525 test images within the dataset. An example of this dataset is shown in Figure 1.

Fig. 1. Example of an image and segmentation mask for the Cityscapes dataset.

B. Data Augmentation

In order to match with [11], before training, each image in the training dataset is augmented. First, the images are randomly scaled between 0.5 and 2.1. Next, the RGB channels in the images are normalized by (102.9801, 115.9465, 122.7717). Then, the images are randomly cropped to a resolution of 512 by 512. Lastly, the images are randomly flipped across the vertical axis. As for the validation set, which is used for evaluation, the only data augmentation is the same normalization applied to the training dataset.

C. Model Architecture

For the architecture of the model, we use the deep neural network model presented in [11]. This model is the DeepLabv3 model detailed in [14]. The input of the model is an image x which is processed by a ResNet101 [15] module resulting in a feature map. The feature map is then passed through an Atrous Spatial Pyramid Pooling (ASPP) [16] module which classifies each pixel. The resulting output of the model is 1/8 of the original resolution, thus, upscaling with bilinear interpolation is applied to the output. The final output of the model is a tensor of size (batch size, height, width, number of classes). The aforementioned architecture is used when training ensembles while an alternate version is employed when training for MC-dropout. The adjusted version for MC-dropout places dropout layers with $p = 0.5$ following the last four ResNet blocks [11].

D. Training Procedure

For ensembling, each model is trained with the hyperparameters shown in Section V. There are a total of 26 models, each trained with the same hyperparameters. To note, though bootstrapping is not used, each model does train on an augmented training dataset due to the randomness introduced by Sub-section III-B. The training procedure for MC-dropout trains only 8 models. MC-dropout uses the architecture that includes the dropout layers.

E. Evaluation

There are three metrics used for evaluation. The first metric is *Area Under the Sparsification Error curve* (AUSE) [17]. AUSE quantifies the sparsification error with a single value, measuring how much the estimated predictive uncertainties coincide with the true prediction errors [11]. The second metric is the *Expected Calibration Error* (ECE) [18]. This metric measures the calibration of the model, indicating whether the model is over-or-under confident in its classifications. The final metric is the *Mean Intersection Over Union* (mIoU) [19]. mIoU quantifies the ability of the model to correctly classify pixels into the correct semantic class.

After training, both ensembling and MC-dropout methods are evaluated according to the evaluation completed in [11]. For ensembling, 8 ensembles of models are randomly chosen (without replacement) for sizes $M \in \{1, 2, 4, 8, 16\}$. The mean and standard deviation of the 8 ensembles' metrics are calculated for each M. When evaluating MC-dropout, the number of forward passes through each model is $M \in \{1, 2, 4, 8, 16\}$. Once again, the mean and standard deviation of the 8 models' metrics are calculated for each M.

IV. RESULTS

The conversion of the code in [11] is still on-going as we are currently ensuring that our implementation behaves as expected and that the two implementations match in terms of the resulting AUSE, ECE, and mIoU. At this time, we only have preliminary results.

In Figure 2, we see that the MC-dropout model performs strongly in the classification of cars, roads, buildings, and vegetation. However, the predicted segmentation image demonstrates the model's difficulty in classifying more minute objects, such as road signs and humans. In terms of entropy, the model's uncertainty is minimized where multiple pixels are connected that belong to the same class, like cars. The uncertainty is maximized in pixels pertaining to boundaries between multiple classes.

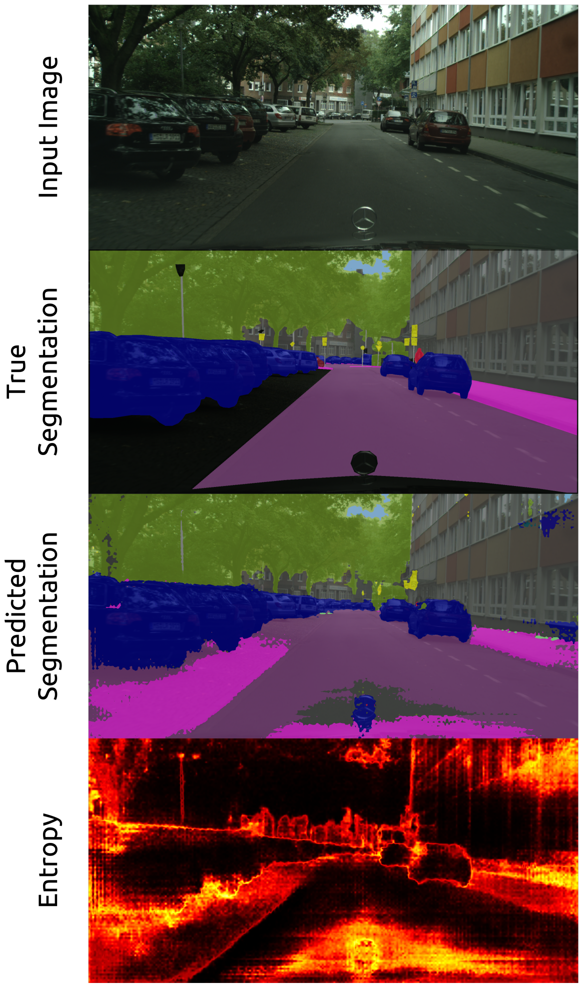


Fig. 2. Preliminary result showing the input image which is passed to a trained MC-dropout model, the true segmentation of the input image, the predicted segmentation output by the model, and the entropy between the true segmentation and the predicted segmentation. In the entropy image, the black pixels represent minimal uncertainty while white pixels represent maximum uncertainty.

Figure 3 allows us to further analyze the segmentation performance of the model. We confirm that the model misclassifies pixels as buildings and cars most frequently. Moreover, sidewalks and roads are often misclassified for one another. This misclassification can be attributed to how sidewalks and roads appear similar to one another and frequently beside one another.

A. Issues and Limitations

While converting the PyTorch implementation in [11] to TensorFlow, there has been numerous conversion problems and lessons to be learned. Firstly, the original architecture makes use of In-Place Activated BatchNorm [20] layers which has its original implementation in PyTorch. Thus, our TensorFlow implementation replaced these layers with a batch normalization layer followed by a LeakyReLU activation layer. Secondly, [11] makes use of PyTorch's CrossEntropyLoss, which has both class weighting and an ignore index as parameters. There is no direct conversion

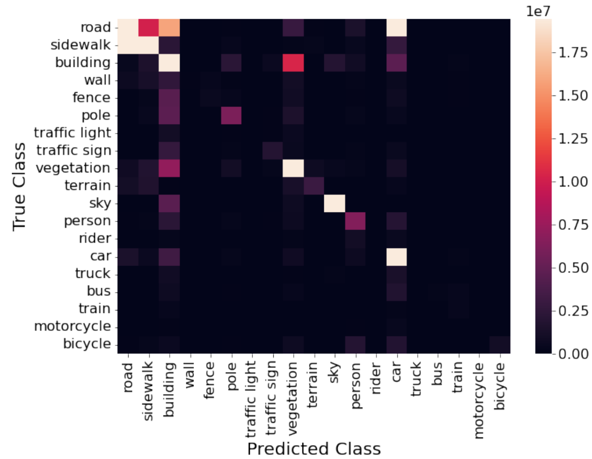


Fig. 3. Confusion matrix heatmap for a single MC-dropout model evaluated on the validation dataset.

of this in TensorFlow, so we manually perform the loss calculation while performing class weighting and using an ignore index. Another issue that arose was that the original implementation uses no specific early stopping mechanism and only trains for a set number of steps. This has resulted in our implementation converting their number of training steps to an approximate number of epochs. Through our calculations, it was determined that training should last for 162 epochs, however, our results have shown that this results in odd behavior in the loss curves occurring around epoch 120. Lastly, [11] uses the train split for training and the validation split for evaluation. This restricts us from determining whether the model is overfitting or underfitting. These aforementioned issues have impeded our progress but we have future steps to address these issues.

V. Conclusion and Future Work

By extending Google Research's UQ framework to include image segmentation, we enable further research on this topic to be explored in the future. Moreover, we provide modular code that can be easily modified to use different models, UQ methods, datasets, and evaluation metrics. We are continuing our work on the code conversion at this time. A thorough review and juxtaposition of the two implementations is planned to determine the root of our current issues. Once our evaluation metrics match with the results in [11], we will use this code base to implement various models and methods.

Further work using this research will focus on anomaly detection. We plan to perform anomaly detection on traffic camera images in order to identify car accidents. Specifically, we will use trajectory images from a novel traffic camera dataset in order to encode multiple video frames into a single image. Through the use of image segmentation as a proxy task, we can perform anomaly detection via uncertainty estimation, outlier exposure, or image re-synthesis [21]. Such anomaly detection on car accidents could determine the safety of specific areas while tracking minor accidents that go unreported.

APPENDIX

Hyperparameters:

- Learning rate: 0.01
- Optimizer: SGD with weighted decay
  - Weight decay: 0.0005
  - Momentum: 0.9
- Crop size: (512, 512)
- Number of classes: 19
- Batch Size: 8 (across 2 NVIDIA RTX A6000 GPUs)
- Training epochs: 81
- Class weighting: 0.8373, 0.918, 0.866, 1.0345, 1.0166, 0.9969, 0.9754, 1.0489, 0.8786, 1.0023, 0.9539, 0.9843, 1.1116, 0.9037, 1.0865, 1.0955, 1.0865, 1.1529, 1.0507

ACKNOWLEDGMENT

The authors would like to thank the RISS community at Carnegie Mellon University, specifically Rachel Burcin and John M. Dolan. Additionally, a special thanks to members of the Auton Lab for their help throughout the RISS program. This work was partially funded by the NSF award 2038612.

REFERENCES

[1] Z. W. Lim, M. L. Lee, W. Hsu, and T. Y. Wong, "Building trust in deep learning system towards automated disease detection," vol. 33, no. 1, pp. 9516–9521. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/5009

[2] R. Michelmore, M. Wicker, L. Laurenti, L. Cardelli, Y. Gal, and M. Kwiatkowska, "Uncertainty quantification with statistical guarantees in end-to-end autonomous driving control," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7344–7350.

[3] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. X. Zhu, "A survey of uncertainty in deep neural networks." [Online]. Available: http://arxiv.org/abs/2107.03342

[4] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," vol. 76, pp. 243–297. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253521001081

[5] Z. Nado, N. Band, M. Collier, J. Djolonga, M. W. Dusenberry, S. Farquhar, Q. Feng, A. Filos, M. Havasi, R. Jenatton, G. Jerfel, J. Liu, Z. Mariet, J. Nixon, S. Padhy, J. Ren, T. G. J. Rudner, F. Sbahi, Y. Wen, F. Wenzel, K. Murphy, D. Sculley, B. Lakshminarayanan, J. Snoek, Y. Gal, and D. Tran, "Uncertainty baselines: Benchmarks for uncertainty & robustness in deep learning." [Online]. Available: http://arxiv.org/abs/2106.04015

[6] K. Wickstrøm, M. Kampffmeyer, and R. Jenssen, "Uncertainty and interpretability in convolutional neural networks for semantic segmentation of colorectal polyps," vol. 60, p. 101619. [Online]. Available: http://arxiv.org/abs/1807.10584

[7] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning." [Online]. Available: http://arxiv.org/abs/1506.02142

[8] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc. [Online]. Available: https://papers.nips.cc/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html

[9] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," p. 10.

[10] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

[11] F. K. Gustafsson, M. Danelljan, and T. B. Schön, "Evaluating scalable bayesian deep learning methods for robust computer vision." [Online]. Available: http://arxiv.org/abs/1906.01620

[12] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[13] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 3213–3223. [Online]. Available: http://ieeexplore.ieee.org/document/7780719/

[14] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation." [Online]. Available: http://arxiv.org/abs/1706.05587

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 770–778. [Online]. Available: http://ieeexplore.ieee.org/document/7780459/

[16] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs." [Online]. Available: http://arxiv.org/abs/1606.00915

[17] E. Ilg, Çiçek, S. Galesso, A. Klein, O. Makansi, F. Hutter, and T. Brox, "Uncertainty estimates and multi-hypotheses networks for optical flow." [Online]. Available: http://arxiv.org/abs/1802.07095

[18] M. Pakdaman Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," vol. 29. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/9602

[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation." [Online]. Available: http://arxiv.org/abs/1411.4038

[20] S. R. Bulò, L. Porzi, and P. Kontschieder, "In-place activated BatchNorm for memory-optimized training of DNNs." [Online]. Available: http://arxiv.org/abs/1712.02616

[21] G. Di Biase, H. Blum, R. Siegwart, and C. Cadena, "Pixel-wise anomaly detection in complex driving scenes." [Online]. Available: http://arxiv.org/abs/2103.05445

Causality-Based Estimation of Traffic Interaction Dynamics

Caleb Chang[1], Yiwei Lyu[2] and John M. Dolan[3]

***Abstract*— Model-based safety-critical control techniques such as Control Barrier Functions (CBF) enable autonomous vehicles to avoid multiple objects safely. A recent application of CBF is the Safe Adaptive Merging Algorithm using Parametric-CBF—a prediction-based approach that enables the ego vehicle to efficiently merge lanes with another vehicle (the prediction vehicle) given its driving behavior derived from past observations. However, the algorithm assumes that pairwise interactions between the prediction vehicle and other surrounding vehicles are already known and that all observations are noiseless. This paper relaxes these assumptions by 1) applying the Cross Map Smoothness (CMS) algorithm in an autonomous vehicle context and 2) developing a robust parameter estimation (RPE) algorithm insusceptible to noise. We take advantage of the CBF controller to identify the causal relationship between the prediction vehicle's acceleration and the CBF of each surrounding vehicle using CMS at each time step. After establishing Granger-causality between surrounding vehicles and the prediction vehicle, we perform robust parameter estimation on the prediction vehicle and formulate the ego's parameters to act accordingly.**

I. Introduction

Human drivers and autonomous vehicles are constantly aware of other vehicles on the road. Intuitively, human drivers are able to provide a reason for their actions. For example, if a lead vehicle slows down in front of the human-driven vehicle, the latter will brake due to the sudden reduced distance between the lead vehicle and itself. In other words, the human is able to identify the lead vehicle as the causal source of the human-driven vehicle's reactive behavior. Autonomous vehicles on the other hand try to keep track of every object all the time. While autonomous vehicles actively try to avoid any unsafe situations [1], the vehicle cannot intuitively reason which obstacle is the cause for its reactive behavior. Understanding the cause for an AV's actions can help quantitatively describe the AV's safety behavior.

CBF-based methods [2] have gained increasing popularity due to their forward-invariant property that provides a theoretical safety-guarantee. A less popularized property of CBF formulation is the adjustable weight in its quadratic programming constraint. This weight or parameter helps describe the admissible control space that the AV can act in [2]- [3]. Essentially, it describes what safety-actions the AV is allowed to take as it approaches an unsafe state, thereby describing the safety behavior of the autonomous vehicle. Typically, this parameter is chosen by the designers of the AV [3] to prioritize safer actions over aggressive ones. To test the effectiveness of the safe controller, a ramp merging scenario is typically used. In the case where multiple cars are on one side of the ramp and the autonomous vehicle on the other, the AV will wait until all other cars have passed before merging to maximize the safety of the scenario. However, in the case of constant traffic on the ramp, the AV will never be able to merge, since its control space is too conservative. To address this, the autonomous vehicle should be less conservative and squeeze its way between other vehicles. We aim to achieve this by learning the safety-behaviors of other vehicles on the ramp to determine which pair of vehicles is the safest to merge between. Our **main contributions** are: 1) introducing a causality identification algorithm into an autonomous vehicle context to establish pairwise-vehicle interactions and 2) developing a novel robust parameter estimation algorithm to learn the safety-behavior of vehicles.

[1]The author is a visiting researcher from the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA. Email: calebwychang@gatech.edu

[2]The author is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213 USA. Email: yiweilyu@andrew.cmu.edu

[3]The author is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213 USA. Email: jdolan@andrew.cmu.edu

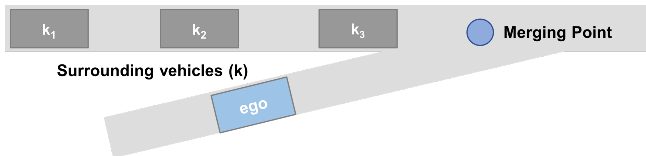


Fig. 1. Ego vehicle seeks to establish pairwise vehicle dynamics and learn α's of all surrounding vehicles. Then, the ego seeks to formulate its own α and control inputs to merge with the other vehicles.

II. Related Works

A primary aspect of autonomous driving is the vehicle's safety functionality [1], [4]. This has sparked many advances in safety critical control. Recently, a popular method to formally guarantee a vehicle's safety is the Control Barrier Function [2]- [5]. In multi-agent systems, Wang et al. and Luo et al. guarantee safe robot swarm behavior [6], [7]. Other non-CBF approaches to safety behavior are presented by van den Berg and Alonso-Mora in Reciprocal Velocity Obstacles [8], [9]. This work primarily focuses on Lyu's Parametric-Control Barrier Functions that provide a richer and more descriptive safety set compared to its vanilla variant by parameterizing the safety function [3].

A great interest of many researchers is system identification for autonomous vehicles. Basic approaches include variants of the Kalman Filter [10], [11]. Grover et al. provides a method to learn parameters of multi-robot systems [12], but no current works focus on parameter identifica-

< TABLE OF CONTENTS

tion of the constraint parameter of CBFs. Lyu provides a simple and efficient optimization to learn the parameters of CBFs under the assumption of zero noise. However, many works introduce noise to evaluate parameter learning in more realistic scenarios [7], [13], [14] so there is a need to learn CBF parameters in uncertain environments. However, one requirement to learn the CBF parameter is that the pairwise-interactions between vehicles need to be known to generate the dataset to learn from [3]. Since not all vehicles are affecting the same vehicle on the road, we need to understand which vehicles cause a change in the behavior of the prediction vehicle. Knowing the causal relationships between vehicles establishes the pairwise vehicle interaction. A widely accepted theory on causality by Pearl [15], [16] uses the notion of interventions to determine cause and effect between variables. Interventions, on a high-level, fix certain variables to a value to determine how the other variables react in response to that fixed-value. Pearl and subsequent researchers use structural equation models (SEM) and graphical approaches to mathematically calculate causality [17]. However, Pearl-causality has limited applicability to scenarios in which the cause and effect variables do not reach an equilibrium (eg. in a Bertrand duopoly two-player gamer) [18]. We turn to Granger-causality for a more practical approach to determining causality.

Granger first introduced his theory in [19]. He used cross-spectral methods and Cramer's representation to formally prove causality. Granger-causality involves probabilistic dependence instead of interventions and functional models as in Pearl-causality [18] and is defined to be the following: "For two simultaneously measured signals, if we can predict the first signal better by using the past information from the second one than by using the information without it, then we call the second signal causal to the first one." [20]. (For the rest of this paper, Granger-causality, will simply be referred to as causality.) Causality has applications to many fields including economics, biology, and ecology. For example, Hesse applies causality to EEG data to understand brain functionality [21]. Bressler et al. uses it to study the general nervous system [22]. Sugihara et al. uses it to study predator-prey relations [23].

Lungarella et al. survey different methods (entropy, regression, similarity index, and predictability improvement techniques) to identify causal relationships, including a way to formulate causality for linear models [24]. A key drawback to most approaches (including the ones just listed) is that they all require large amounts of noise-free data [25]. Later, Sugihara et al. develop the Cross Convergent Mapping (CCM) algorithm to detect causality between different factors in an ecosystem using time-embeddings [23]. However, like the methods mentioned by Lungarella, CCM still requires hundreds to thousands of observations. Another work by Ancona et al. uses Radial Basis Functions to handle nonlinear cases for causality detection [26]. Ma et al. expand on Ancona's and Sugihara's work by introducing the Cross Map Smoothness (CMS) algorithm [27]. Unlike previous approaches that required large amounts of noise free data, CMS only requires a few observations, is robust to noise, and works for nonlinear time-series in dynamical systems. These properties make CMS the ideal algorithm for establishing pairwise vehicle dynamics since vehicle models can be nonlinear and are heavily susceptible to noise in the real world.

III. Methods & Testing

A. Background on Parametric Control Barrier Functions

1) Vanilla Control Barrier Function: Control Barrier Functions (CBF) [5] are used to define an admissible control space for provable safety in dynamical systems. One of the method's important properties is its forward-invariance guarantee of a desired safe set that repels safe states near the boundary back into the safe set. Consider the following nonlinear system in control affine form:

$$\dot{x} = f(x) + g(x)u \tag{1}$$

where $x \in \mathcal{X} \subset \mathbb{R}^n$ and $u \in \mathcal{U} \subset \mathbb{R}^m$ are the system state and control input with f and g assumed to be locally Lipschitz continuous. A desired safe set $x \in \mathcal{H}$ can be denoted by a safety function $h(x)$:

$$\mathcal{H} = \{x \in \mathbb{R}^n : h(x) \geq 0\} \tag{2}$$

Thus, the control barrier function for the system to remain in the safety set can be defined as follows [5]:

Definition 1. (Control Barrier Function) Given a dynamical system (1) and the set $\mathcal{H}$ defined in (2) with a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, then h is a control barrier function (CBF) if there exists a class $\mathcal{K}$ function for all $x \in \mathcal{X}$ such that

$$\sup_{u \in \mathcal{U}} \{L_f h(x) + L_g h(x)u\} \geq -\kappa(h(x)) \tag{3}$$

where $\dot{h}(x,u) = L_f h(x) + L_g h(x)u$ with L_f and L_g as the Lie derivatives of h along the vector fields f and g.

A commonly selected class $\mathcal{K}$ function is $\kappa(h(x)) = \gamma h(x)$ [4] [5], where $\gamma \in \mathbb{R}^{\geq 0}$ is a CBF design parameter controlling system behaviors near the boundary of $h(x) = 0$. Hence, the admissible control space in (3) can be redefined as

$$\mathcal{B}(x) = \{u \in \mathcal{U} : \dot{h}(x,u) + \gamma h(x) \geq 0\} \tag{4}$$

It is proved in [5] that any controller $u \in \mathcal{B}(x)$ will render the safe state set $\mathcal{H}$ forward-invariant, i.e., if the system (1) starts inside the set $\mathcal{H}$ with $x(t = 0) \in \mathcal{H}$, then it implies $x(t) \in \mathcal{H}$ for all $t > 0$ under controller $u \in \mathcal{B}(x)$. However, the particular form $\kappa(h(x)) = \gamma h(x)$ is limited in describing complicated system behaviors when approaching the boundary of $h(x) = 0$. The Parametric-Control Barrier Function expands on the vanilla Control Barrier Function to address this weakness.

2) Parametric Control Barrier Function: Lyu et al. introduce the Parametric-CBF, an extension of the vanilla CBF. They address the limited description disadvantage of the vanilla CBF by changing the mapping function $\kappa(h(x))$ to be a q-degree polynomial [3]:

Definition 2. (Parametric-Control Barrier Function) Given a dynamical system (1) and the set $\mathcal{H}$ defined in (2) with a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, then h is a Parametric-Control Barrier Function (Parametric-CBF) for all $x \in \mathcal{X}$ such that

$$\underset{u \in \mathcal{U}}{sup}\{\dot{h}(x,u)\} + \alpha H \geq 0 \tag{5}$$

where parameter vector $\alpha = [\alpha_1, \alpha_2, ..., \alpha_q] \in \mathbb{R}^q$ with $\forall \alpha_p \in \mathbb{R}^{\geq 0}$ for $p \in [q]$. $H(x) = [h(x), h^3(x), h^5(x), ..., h^{2q-1}(x)]^T, q \in \mathcal{N}$.

The proof of properties of the Parametric-CBF can be found in [5]. The advantages of the Parametric-CBF over its vanilla variant are its richer descriptive information and its broader applications to autonomous vehicles. The polynomial nature of the $\kappa(h(x))$ allows the Parametric-CBF to describe the safety behavior to differing degrees by the linear combination of parameter vector α and the high-order safety measurement vector $H(x)$. In a broader context, machine learning can be used to learn and predict the driving behavior of a Parametric-CBF-based controller by learning the parameter α over time observations. The authors use linear Ridge regression in an ideal noiseless driving scenario.

An important contribution from [3] is the Safe Adaptive Merging Algorithm, which uses Parametric-CBF to safely and efficiently merge with another vehicle. The algorithm is used in a ramp merging scenario which proceeds in the following fashion: 1) A surrounding vehicle (k) and prediction vehicle (j) are interacting on one side of the ramp. 2) The ego vehicle (i) learns the prediction vehicle's α_j based on observations between the surrounding and prediction vehicle from the other side of the ramp. 3) The ego vehicle handcrafts its own α_i to efficiently merge with the prediction vehicle. The presented algorithm has two major assumptions. First, in the event of multiple surrounding vehicles, the ego vehicle assumes it knows which surrounding vehicle the prediction vehicle interacts with. Second, the estimation technique, linear Ridge regression, is highly susceptible to noise. Thus, a method to establish inter-vehicle causality and a robust estimate technique is needed to better exemplify a real driving scenario.

B. Problem Statement

The ramp merging situation is shown in Fig. 1. The goal is to allow the ego vehicle to merge effectively with other vehicles on the opposite side of the ramp with motion uncertainty. The system dynamics can be described as a double integrator as follows:

$$\dot{X} = \begin{bmatrix} 0_{2\times2} & I_{2\times2} \\ 0_{2\times2} & 0_{2\times2} \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0_{2\times2} & I_{2\times2} \\ I_{2\times2} & 0_{2\times2} \end{bmatrix} \begin{bmatrix} u \\ \epsilon \end{bmatrix} \tag{6}$$

where $x \in \mathcal{X} \subset \mathbb{R}^2, v \in \mathbb{R}^2$ are the position and linear velocity of each car and $u \in \mathbb{R}^2$ is the acceleration control input. $\epsilon\ \mathcal{N}(\hat{\epsilon}, \Sigma)$ is a random uniform noise variable with known mean $\hat{\epsilon} \in \mathbb{R}^2$ and variance $\Sigma \in \mathbb{R}^{2\times2}$, representing uncertainty in each vehicle's motion. All vehicles deploy a heterogeneous Parametric-CBF controller and are expected to maintain task efficiency. The objective function can be formulated as a quadratic programming problem for all pairs of vehicles in the scenario:

$$\begin{aligned} &\underset{u_i \in \mathcal{U}_i}{\arg\min} \|u_i - \bar{u}_i\|^2 \\ s.t. \quad &U_i^{min} < u_i < U_i^{max} \\ &\dot{h}_{ij}(x,u) + \alpha H_{ij} \geq 0 \end{aligned} \tag{7}$$

where i and j are the indices of the pairwise vehicles. $\bar{u}_i$ is the nominal acceleration for the i^{th} vehicle to follow; it is assumed to be computed by a higher-level task-related planner. U_i^{max} and U_i^{min} are the maximum and minimum input acceleration limits. We consider the following choice of safety function $h_{ij}(x)$ and safety set $\mathcal{H}_i$.

$$\mathcal{H}_i(x) = \{x \in \mathcal{X} : h_{ij}(x) = \|x_i = x_j\|^2 - R_{safe}^2 \geq 0 \quad \forall i \neq j\} \tag{8}$$

where x_i and x_j are the positions of each pairwise set of vehicles and $R_{safe} \in \mathbb{R}^+$ is the minimum allowed safety distance.

For general convention, we consider the i vehicle to be the ego vehicle whose u_i and α_i can be controlled. We consider the j vehicle to be the prediction vehicle whose α_j is learned by the ego vehicle. We consider the $k_1, \ \ldots \ , k_n$ vehicles to be the n surrounding vehicles which may interact with the prediction vehicle. All vehicles deploy a Parametric-CBF-based controller with different α's.

C. Causality Detection for Pairwise Vehicle Dynamics

An adaptive cruise control (ACC) vehicle that deploys a Control Barrier Function and Control Lyapunov Function is able to accelerate towards its desired velocity permitted that it maintains a safe distance from its lead vehicle [2]. In the event that the ACC vehicle slows down to satisfy the CBF constraint between the lead vehicle and ACC vehicle, we say that the lead vehicle causally influences the ACC vehicle. We introduce the Cross Map Smoothness algorithm [27] as a way to quantify "how much" the behavior of one vehicle influences the behavior of another.

1) Cross Map Smoothness: The algorithm is based on state space reconstruction through time-delayed embeddings [27]. Consider two discrete-time observations $x(t)$ and $y(t)$ in which we seek to understand the influence of x on y, or more succinctly: $x \to y$. Given an embedding dimension L and time delay τ, the time-delayed coordinate vectors are formulated as $\mathbf{x}(t) = [x(t), x(t-\tau), ..., x(t-(L-1)\tau)]^T$ and $\mathbf{y}(t) = [y(t), y(t-\tau), ..., y(t-(L-1)\tau)]^T$. Together, a set of vectors $\mathbf{x}(t)$ form the reconstructed attractor M_x, likewise for M_y. The k-nearest neighbors of a point $\mathbf{y}(t_0)$ on M_y are $\mathbf{y}(t_{y,1}), \mathbf{y}(t_{y,2}), ..., \mathbf{y}(t_{y,k})$. The k-nearest neighbors have corresponding mutual neighbors of $\mathbf{x}(t_0)$ on attractor

M_x: $\mathbf{x}(t_{y,1}), \mathbf{x}(t_{y,2}), ..., \mathbf{x}(t_{y,k})$ for the same points in time. If x causally influences y, there is a smooth cross map Φ_{yx} that maps two close states on the attractor M_y to two corresponding close states on M_x. If x does not causally influence y, the corresponding states are not necessarily close to each other. In other words, there would be no smooth mapping from points on M_y to points on M_x. Further details and explanation can be found in [27].

The central idea of the CMS algorithm according to Ma et al. is the ability of a neural network to represent any smooth map [27]. A Radial Basis Function Neural Network (RBFNN) is trained on a set of data $\mathbf{y}(t)$ as the input and the corresponding set of data $\boldsymbol{x}(t)$ as the output. The training error of the RBFNN is reflective of the smoothness of the cross mapping Φ_{yx}. A small error indicates a smooth mapping, which indicates a causal influence of x on y. Likewise, a large error indicates a rough mapping, which indicates no causality between the two variables. Additionally, a finding by Ma et al. is that the algorithm is generally robust to noise. The algorithm is presented below.

Algorithm 1: Cross Map Smoothness Algorithm

```
Data: x_1, x_2, ..., x_n ∈ ℝ^L and y_1, y_2, ..., y_n ∈ ℝ^L,
      let S_i = {1, 2, ..., n}\i be the leave-one-out
      index set.
Result: Causality Index: R_xy ∈ [0, 1]
for i = 1, 2, ..., n do
    train radial basis function neural network N_i
      based on N_i(y_j) = x_j, j ∈ S_i
    x̂_i ← N_i(y_j)
    ε_i ← ||x_i − x̂_i||
Normalize the error: Δ ← rms(ε) / rms(||x − x̄||)
/* σ is a positive constant used to
   normalize R_xy. σ = 5 is used.       */
Causality Index: R_xy ← 1 / exp(Δ/σ)
```

2) Autonomous Vehicle Context: To determine the causal influence of one vehicle on another, we apply the CMS algorithm in an autonomous vehicle (AV) context. There are many AV-specific candidate variables that could be used as inputs $x(t)$ and $y(t)$ to the algorithm: position, velocity, acceleration, distance from other vehicles, etc. In this paper, we choose the CBF safety function as $x(t)$ and the norm of acceleration as $y(t)$, which can be expressed as $h_{jk} \rightarrow acc_j$ for the prediction vehicle and surrounding vehicle. Since acceleration is the control input (u_j), the expression describes how the safety measurement between two vehicles causally influences the action of the prediction vehicle. The output of the CMS algorithm R_{xy} is a number ranging from 0 to 1. If $x \rightarrow y$, then $R > 0$. If not, then $R = 0$ or $R \simeq 0$. The value of R represents to a relative degree how causal x is [27]. However, since we merely want to establish whether or not one vehicle is influencing another, we treat R as more of a binary term and say that there is a significant interaction if R is greater than some threshold z. Also, in the event that the prediction vehicle is interacting with multiple surrounding vehicles, only one dynamic needs to be established, since parameter estimation only relies on one set of interaction observations between vehicles. It is easiest to choose the surrounding vehicle with the most significant interaction (indicated by the highest peak in R) as part of the pair with the prediction vehicle.

Remark 1. When the CBF-QP constraint is active, $h(x) = 0$ and $\dot{h}(x, u) = 0$, thereby causing $R = 0$ since there is no change in $h(x)$ over time. Simulation results show that when two vehicles start to interact, there is an increase in R. When the constraint is finally active, R drops to 0. As the constraint deactivates, R increases again before dropping off again to 0 to indicate no causality. The R plot of a vehicle activating and deactivating its constraint looks like a 'M' with the constraint activation period surrounded by two peaks (one for activation and the other for deactivation).

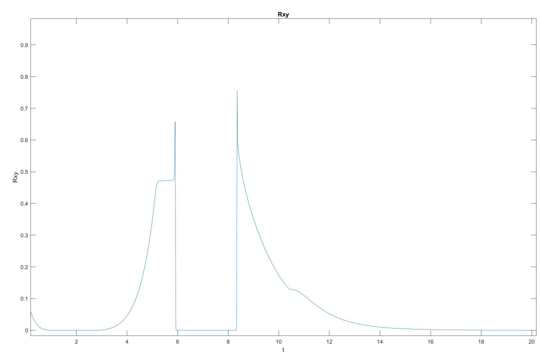


Fig. 2. Adaptive Cruise Control Example. The lead vehicle slows down $t \simeq 0 - 6s$; the constraint starts to activate. When the constraint is active ($t \simeq 6 - 8.2s$), the ACC vehicle maintains its safety distance and $R = 0$. At $t \simeq 8.2s$, the lead vehicle significantly speeds up, allowing the ACC vehicle to pursue its desired velocity; the constraint deactivates. The overall 'M' figure can be roughly seen as described in *Remark 1*.

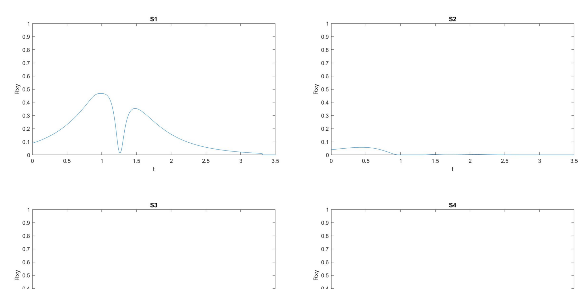


Fig. 3. Multi-Agent Example. The autonomous vehicle first interacts with the fourth surrounding vehicle (S4) and then interacts with the first surrounding vehicle (S1). It never interacts with the second or third surrounding vehicle. We observe the first dip between two peaks in S4 and the second dip in S1 in the $R(t)$ curves. As expected, no causality is detected for S2 or S3.

< TABLE OF CONTENTS

Remark 2. Causality is not automatically detected by the CMS algorithm. It may need some time to converge, as seen in the first few time steps of the simulation in Figures 2 and 3.

D. Behavior Prediction

Once pairwise vehicle dynamics have been established, regression techniques can be used to learn or estimate the α of the prediction object. Since α describes the safety behavior of the vehicle, it can only be learned when the safety constraint is active. In other words, α can only be learned during periods of causality between two vehicles. Two parameter learning methods are presented for a noiseless and noisy case.

1) Noiseless Parameter Estimation: Lyu et al. is able to learn the parameters of the Parametric-CBF in [3]. Their method utilizes Ridge linear regression to learn α_j with recorded errors in the 10^{-6} range. In the noiseless case, another way to learn α is through a modified version of least squares regression:

$$\bar{\alpha}_j = \arg\min_{\alpha_j} \sum_{t=1}^{m} ||\dot{h}_{jk}^t - \alpha_j H_{jk}^t||_2^2 \tag{9}$$

Instead of posing parameter estimation as an optimization problem, we can pose it as an inverse problem. Taking the derivative of the error term $E = \sum_{t=1}^{m}(\dot{h}_{jk}^t - \alpha_j H_{jk}^t)^2$ with respect to α, we can simplify the result into $\mathbb{A}_{sum}\bar{\alpha}_j = \mathbb{B}_{sum}$, where $\mathbb{A}_{sum}$ and $\mathbb{B}_{sum}$ are summations of $\mathbb{A}^t$ and $\mathbb{B}^t$ over time. We solve for $\bar{\alpha}_j$ by taking the inverse of $\mathbb{A}_{sum}$:

$$\bar{\alpha}_j = [-\mathbb{A}_{sum}^{-1}\mathbb{B}_{sum}]^T \tag{10}$$

$\mathbb{A}^t$ and $\mathbb{B}^t$ can be calculated online from observations, which allows for real-time parameter estimation.

$$\begin{aligned} \mathbb{A}^t &= H_{jk}^t {H_{jk}^t}^T \\ \mathbb{B}^t &= H_{jk}^t \dot{h}_{jk} \end{aligned} \tag{11}$$

Similar to [3], we wait for α_j to converge before returning the learned parameter. However, in the presence of noise, α_j will not converge.

2) Robust Parameter Estimation and Applications: We introduce a novel heuristic check in addition to the required convergence for predicting α_j. In an ideal world where our estimated $\bar{\alpha}_j$ perfectly matches α_j, the CBF constraint is satisfied such that $A_j u_j - b_j = 0$ where $A_j = -2{\Delta x_{jk}^t}^T$ and $b_j = 2{\Delta x_{jk}^t}^T(\Delta v_{jk}^t - u_k \Delta t) + \alpha_j H_{jk}^t$. Therefore, if $\bar{\alpha}_j \simeq \alpha_j$, then $\bar{A}_j u_j - \bar{b}_j \simeq 0$ should also be true, where $\bar{A}_j$ and $\bar{b}_j$ are estimated CBF-QP constraints based on $\bar{\alpha}_j$. We have α_{new} as the newly estimated $\bar{\alpha}_j$:

$$\bar{A}_j = A_j = -2{\Delta x_{jk}^t}^T \tag{12}$$

$$\bar{b}_j = 2{\Delta x_{jk}^t}^T(\Delta v_{jk}^t - u_k \Delta t) + \alpha_{new} H_{jk}^t \tag{13}$$

where $\Delta x_{jk} = x_j - x_k$ and $\Delta v_{jk} = v_j - v_k$. The Robust Parameter Estimation (RPE) algorithm checks for this property when estimating $\bar{\alpha}_j$. Given that this is a heuristic check, there are some estimated values that are not correct. Thus, we add all estimated values into a dataset $\mathbb{D}$ and perform k-means clustering as a sort of voting method to remove any inaccurate estimations. However, if $\bar{A}_j u_j - \bar{b}_j$ is not close enough to 0, we restart the summation of $\mathbb{A}$ and $\mathbb{B}$ and start recalculating a new estimate.

Remark 3. Calculating the true dot derivative of the dynamics presented in Eq. 6 actually results in a different b_j formulation than the one presented in Eq. 13:

$$b_j = {\Delta x_{jk}^t}^T(\Delta\epsilon_{jk} + \Delta v_{jk}^t - u_k \Delta t) + \alpha_{new} H_{jk}^t \tag{14}$$

The addition of the error term: $\Delta\epsilon_{jk} = \epsilon_j - \epsilon_k \sim \mathcal{N}(\Delta\hat{\epsilon}_{jk}, \Delta\Sigma_{jk})$ is ignored when calculating the estimation constraints. Instead, the noise is thought to be already built into the dynamics, so the addition of the noise term is not necessary when formulating constraints.

Algorithm 2: Robust Parameter Estimation

Data: $\Delta x_{jk}, \Delta v_{jk}, u_j, u_k, \Delta t, R_{safe}$
Result: $\bar{\alpha}_j$
Initialize $\mathbb{A}_{sum}$ and $\mathbb{B}_{sum}$ to 0
for $t = 1 : m$ **do**
    Calculate $\mathbb{A}^t$ and $\mathbb{B}^t$ with (11)
    Add $\mathbb{A}^t$ and $\mathbb{B}^t$ to their respective summations $\mathbb{A}_{sum}$ and $\mathbb{B}_{sum}$
    Calculate α_{new} with (10)
    Calculate estimated constraints $\bar{A}_j$ and $\bar{b}_j$ with (12-13)
    $c \leftarrow \bar{A}_j u_j - \bar{b}_j$
    $\epsilon \leftarrow \texttt{rmse}(\alpha_{new}, \alpha_{old})$
    if $c < \delta_c$ **then**
        Reinitialize $\mathbb{A}_{sum}$ and $\mathbb{B}_{sum}$ to $\mathbb{A}^t$ and $\mathbb{B}^t$
    else if $c \leq 0$ *and* $\epsilon < \delta_{rmse}$ **then**
        add α_{new} to dataset $\mathbb{D}$
    $\alpha_{old} \leftarrow \alpha_{new}$
Perform k-means clustering on $\mathbb{D}$
$\bar{\alpha}_j \leftarrow$ average of the most populated cluster

The parameter estimation task is validated on 30 trials with randomly generated driving styles with $\alpha_j \in \mathbb{R}^2$ and $\hat{\epsilon} = 0$ and $\Sigma = 10$. Typical works such as [13] use zero-mean noise with standard deviations less than 1, but we specifically chose a much higher standard deviation to demonstrate robustness. Also note that $\delta_c \in \mathbb{R}^-$ and $\delta_{rmse} \in \mathbb{R}^+$ are hyperparameters that need to be tuned. The closer these hyperparameters are to zero, the stricter the heuristic check is. Effectively, more accurate estimates of α_j are found at the cost of finding them less often. If the hyperparameters are too close to 0, there is a chance that no estimates are added to $\mathbb{D}$, resulting in no $\bar{\alpha}_j$ returning from the RPE algorithm. For these trials, $\delta_c = -10^{-2}$ and $\delta_{rmse} = 10^{-4}$. The average rmse between the true value and estimated value across all 30 trials is 2.35×10^{-3}. A notable performance observation occurs when we remove the top 3 most inaccurate observations: the average rmse across the remaining 27 trials drops 3 orders of magnitude lower to 8.93×10^{-6}. The same trials are rerun under Gaussian noise ($\hat{\epsilon} = 0$ and $\Sigma = 1$) for a more standard scenario as in [3]; the average rsme is 2.34×10^{-3}. A histogram of the errors is shown in Fig. 4. Regardless

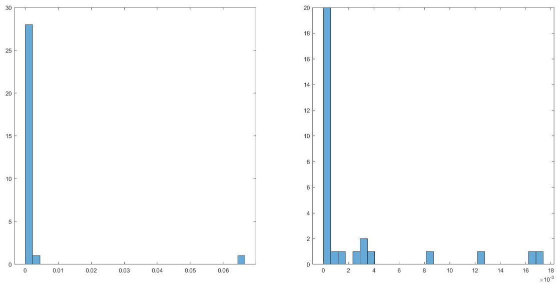


Fig. 4. RPE is tested under 30 trials under random α initializations for both cases. Regardless of the noise, most of the rmse are concentrated in the smallest bucket. Left: Histogram of average rsme between true and estimated α with uniform noise. Right: Same histogram but with Gaussian noise instead.

of which noise is considered, it has been demonstrated that the RPE algorithm performs extremely well in the presence of noisy dynamics. Even higher accuracies can be achieved with filtering techniques such as a Kalman Filter [10], [11]. Overall, the same conclusion from [3] can be made: this Parametric-CBF-based prediction framework does not lose its generality by achieving a consistent prediction performance.

E. Robust Multi-Agent Safe Adaptive Merging control Algorithm using Parametric-CBF

We present an extension of the Safe Adaptive Merging Algorithm that allows for the ego vehicle to merge with multiple heterogeneous robots in uncertain environments. In Algorithm 2, we distinctly separate the prediction vehicle from the surrounding vehicle. However, with Algorithm 3, we treat all vehicles (except the ego) on the ramp to be the prediction vehicle and all others with respect to that vehicle to be the surrounding vehicles. Then, the causality index R can be calculated for each vehicle at the same time. If a causal relationship has been established given a vehicle and its surrounding vehicle, we estimate its parameters with RPE. After m time steps, we assume the parameters for all vehicles have been learned: the overall system has been identified. We handcraft our own α_i for more efficient merges. Lyu et al. suggests in [3] that the general tuning strategy to avoid potential conflicts in safe-driving behavior between vehicles is for the ego to demonstrate the opposite behavior of the prediction vehicle. If the prediction vehicle is very aggressive, the ego vehicle should be more defensive and vice versa.

Algorithm 3: Robust Multi-Agent Safe Adaptive Merging control Algorithm using Parametric-CBF

Data: $\Delta x_{ij}, \Delta x_{jk_1}, ..., \Delta x_{jk_n}, \Delta v_{ij}, \Delta v_{jk_1}, ..., \Delta v_{jk_n}, \Delta t, R_{safe}$

Result: $\bar{\alpha}_j, u_i$

Calculate controls inputs: $u_j, u_{k_1}, ..., u_{k_n}$ from velocity observations

for $t = 1 : m$ **do**

  Perform $CrossMapSmoothness$ concurrently for all vehicles to determine vehicle interactions over time

  Perform $RobustParameterEstimation$ concurrently for all vehicles with active constraints

Choose the appropriate α_i based on $\bar{\alpha}_{k_1}, ..., \bar{\alpha}_{k_n}$

for *t=m:N* **do**

  Compute safety constraint parameter A_{ij}^t and b_{ij}^t

  $u_i^t = \underset{u_i \in \mathcal{U}_i}{\arg\min} \|u_i - \bar{u}_i\|^2$ with constraints in (7)

IV. Conclusion

In this work, we expand upon the Safe Adaptive Merging Algorithm by relaxing two key assumptions: vehicle pairwise-interactions are known and the dynamics are noise-free. We introduce the Cross Map Smoothness algorithm, which is typically applied in fields like biology and ecology, into an autonomous vehicle context to establish vehicle pairwise-interactions. Then, we develop a novel heuristic-based approach called Robust Parameter Estimation to learn the safety-behavior of vehicles.

A. Future Work

Currently, RPE demonstrates promising experimental results but it is unknown why its heuristic is so effective especially since the estimated constraints completely disregard the noise term. In future work, we would like to develop a more rigorous understanding of RPE to fully justify its robustness in the presence of noise. Second, the safety-behavior parameter α is still a handcrafted value. We would like to find an optimal α that allows for the most efficient merge in multi-agent scenarios.

Acknowledgment

The author would like to express a special thank you to Yiwei Lyu, John M. Dolan, Rachel Burcin, the National Science Foundation and the RISS 2022 cohort for an amazing research experience at Carnegie Mellon. Also, the authors would like to thank Dr. Huanfei Ma for discussions regarding the Cross Map Smoothness algorithm and the CMS toy example he provided.

References

[1] Tianyu Gu, J. Atwood, Chiyu Dong, J. M. Dolan, and Jin-Woo Lee, "Tunable and stable real-time trajectory planning for urban autonomous driving," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Hamburg, Germany: IEEE, Sept. 2015, pp. 250–256. [Online]. Available: http://ieeexplore.ieee.org/document/7353382/

[2] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control Barrier Function Based Quadratic Programs for Safety Critical Systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017, conference Name: IEEE Transactions on Automatic Control.

[3] Y. Lyu, W. Luo, and J. M. Dolan, "Adaptive Safe Merging Control for Heterogeneous Autonomous Vehicles using Parametric Control Barrier Functions," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, June 2022, pp. 542–547.

[4] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *53rd IEEE Conference on Decision and Control*, Dec. 2014, pp. 6271–6278, iSSN: 0191-2216.

[5] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control Barrier Functions: Theory and Applications," in *2019 18th European Control Conference (ECC)*, June 2019, pp. 3420–3431.

[6] L. Wang, A. Ames, and M. Egerstedt, "Safety barrier certificates for heterogeneous multi-robot systems," in *2016 American Control Conference (ACC)*, July 2016, pp. 5213–5218, iSSN: 2378-5861.

[7] W. Luo, W. Sun, and A. Kapoor, "Multi-Robot Collision Avoidance under Uncertainty with Probabilistic Safety Barrier Certificates," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 372–383.

[8] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal Velocity Obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 1928–1935, iSSN: 1050-4729.

[9] J. Alonso-Mora, A. Breitenmoser, P. Beardsley, and R. Siegwart, "Reciprocal collision avoidance for multiple car-like robots," in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 360–366, iSSN: 1050-4729.

[10] E. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, Oct. 2000, pp. 153–158.

[11] R. Van der Merwe and E. Wan, "The square-root unscented Kalman filter for state and parameter-estimation," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 6, May 2001, pp. 3461–3464 vol.6, iSSN: 1520-6149.

[12] J. S. Grover, C. Liu, and K. Sycara, "Parameter Identification for Multirobot Systems Using Optimization Based Controllers (Extended Version)," Sept. 2020, arXiv:2009.13817 [cs, math]. [Online]. Available: http://arxiv.org/abs/2009.13817

[13] Y. Lyu, W. Luo, and J. M. Dolan, "Probabilistic Safety-Assured Adaptive Merging Control for Autonomous Vehicles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 10 764–10 770, iSSN: 2577-087X.

[14] G. Angeris, K. Shah, and M. Schwager, "Fast Reciprocal Collision Avoidance Under Measurement Uncertainty," May 2019, arXiv:1905.12875 [cs, math]. [Online]. Available: http://arxiv.org/abs/1905.12875

[15] J. Pearl, *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Mar. 2000, google-Books-ID: wnGU_TsW3BQC.

[16] ——, "Causal inference," in *Proceedings of Workshop on Causality: Objectives and Assessment at NIPS 2008*, ser. Proceedings of Machine Learning Research, I. Guyon, D. Janzing, and B. Schölkopf, Eds., vol. 6. Whistler, Canada: PMLR, 12 Dec 2010, pp. 39–58. [Online]. Available: https://proceedings.mlr.press/v6/pearl10a.html

[17] P. Spirtes, "Introduction to Causal Inference," p. 20.

[18] H. White, K. Chalak, and X. Lu, "Linking Granger Causality and the Pearl Causal Model with Settable Systems," in *Proceedings of the Neural Information Processing Systems Mini-Symposium on Causality in Time Series*. PMLR, Mar. 2011, pp. 1–29, iSSN: 1938-7228. [Online]. Available: https://proceedings.mlr.press/v12/white11.html

[19] C. W. J. Granger, "Investigating Causal Relations by Econometric Models and Cross-spectral Methods," *Econometrica*, vol. 37, no. 3, pp. 424–438, 1969, publisher: [Wiley, Econometric Society]. [Online]. Available: https://www.jstor.org/stable/1912791

[20] H. Ma, S. Leng, and L. Chen, "Data-based prediction and causality inference of nonlinear dynamics," *Science China Mathematics*, vol. 61, no. 3, pp. 403–420, Mar. 2018. [Online]. Available: https://doi.org/10.1007/s11425-017-9177-0

[21] W. Hesse, E. Möller, M. Arnold, and B. Schack, "The use of time-variant EEG Granger causality for inspecting directed interdependencies of neural assemblies," *Journal of Neuroscience Methods*, vol. 124, no. 1, pp. 27–44, Mar. 2003.

[22] S. L. Bressler and A. K. Seth, "Wiener–Granger Causality: A well established methodology," *NeuroImage*, vol. 58, no. 2, pp. 323–329, Sept. 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053811910002272

[23] G. Sugihara, R. May, H. Ye, C.-h. Hsieh, E. Deyle, M. Fogarty, and S. Munch, "Detecting Causality in Complex Ecosystems," *Science*, vol. 338, no. 6106, pp. 496–500, Oct. 2012. [Online]. Available: https://www.science.org/doi/10.1126/science.1227079

[24] M. Lungarella, K. Ishiguro, Y. Kuniyoshi, and N. Otsu, "Methods for quantifying the causal structure of bivariate time series." *I. J. Bifurcation and Chaos*, vol. 17, pp. 903–921, 03 2007.

[25] S. Sinha and U. Vaidya, "On Data-Driven Computation of Information Transfer for Causal Inference in Discrete-Time Dynamical Systems," *Journal of Nonlinear Science*, vol. 30, no. 4, pp. 1651–1676, Aug. 2020. [Online]. Available: https://doi.org/10.1007/s00332-020-09620-1

[26] N. Ancona, D. Marinazzo, and S. Stramaglia, "Radial basis function approach to nonlinear Granger causality of time series," *Physical Review E*, vol. 70, no. 5, p. 056221, Nov. 2004, publisher: American Physical Society. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.70.056221

[27] H. Ma, K. Aihara, and L. Chen, "Detecting Causality from Nonlinear Dynamics with Short-term Time Series," *Scientific Reports*, vol. 4, no. 1, p. 7464, Dec. 2014, number: 1 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/srep07464

AirMVS: Hierarchical Image Decomposition with Sparse Convolution for Fast Multiview Stereo Vision with Fisheye Images

Conner Pulling[1], Yaoyu Hu[2], and Sebastian Scherer[3]

Abstract— **Depth perception using stereo vision is a vital part of robotic autonomy stacks for many fields such as manipulation, 3D reconstruction, and field robotics. For aerial robotics, stereo vision methods must be real-time and omnidirectional to avoid collisions in all directions while flying at fast speeds. One method to achieve omnidirectional stereo vision is the use of multiple wide field-of-view (FOV) fisheye camera images. Previous methods have shown that learning-based methods work well for this task but are not real time. Additionally, while previous non-learning methods are real-time, they cannot take advantage of learned contextual information in images and rely on photometric parallax effects alone. This work presents a novel learning-based approach to real-time multiview omnidirectional stereo vision that achieves state-of-the-art (SOTA) accuracy and resolution. Using image decomposition and sparse convolution, resolution scales are separately processed and then recombined. This method preserves high spatial frequency features and reduces redundancies amongst low-spatial frequency features without unnecessary computation. Additioanlly, this work presents a synthetic multiview dataset of fisheye images with approximately 9,000 samples made with a novel flexible data collection pipeline.**

I. INTRODUCTION

Depth Perception is an incredibly important capability for humans and robots alike, giving agents important information about how to navigate and manipulate their surroundings. When determining how far to reach to pick up a cup or navigate around obstacles, depth information is essential. Many prior works in fields such as agriculture robotics, 3D reconstruction, and aerial robotics often use depth information in addition to RGB images as input to their methods [1] [2] . Improving depth perception methods leads to better performance on these downstream tasks and reduces the propagation of error. Thus, having robust and accurate depth perception methods provides a great foundation for research that relies on depth information. Additionally, these methods must be as fast as possible to better justify their use in real world field robotics tasks.

This need for computationally efficient and fast depth estimation algorithms is especially present for Unmanned Aerial Vehicles (UAVs). UAVs, unlike many other robotic systems, move three-dimensionally and often at high speeds. Thus, UAVs need to employ omnidirectional depth estimation methods so that the UAV simultaneously knows where all obstacles are in the environment. LIDAR is a common solution, however these systems can be heavy, expensive, and often do not work well in dusty or reflective environments. Additionally, LIDARs return sparse point clouds and are not ideal for tasks that require a dense depth map, such as 3D reconstruction. Camera-based stereo vision methods can be less expensive, provide dense depth information, and work better in environments where LIDAR fail. The tradeoff is that typical pinhole cameras lenses cannot have fields-of-view (FOVs) exceeding 180 degrees. Most commercial camera systems do not even have FOVs that exceed 120 degrees. Therefore, omnidirectional camera-based stereo vision might require many pinhole cameras to be functional, increasing hardware needs and computational complexity. However, fisheye lenses can be used to achieve FOVs exceeding 180 degrees with the caveat that the image is heavily distorted. However, learning-based and non-learning methods alike have achieved success on predicting depth from fisheye images [3], [4]. However, real-time inference with current learning-based multiview stereo vision (MVS) methods using multiple fisheye images is still a challenge and requires desktop or higher grade GPUs that would be impractical to use on UAVs [3], [5], [6]. Recent non-learning methods achieve real-time performance and outperform current learning-based methods, yet this method only uses one stereo pair per pixel [4]. Prior works have shown that resolving depth using multiple stereo pairs in pinhole images can better resolve ambiguities that prove challenging for traditional binocular stereo vision [7]. Lastly, No prior learning-based methods have used datasets with large intrasample variety, which raises the concern that using data from a higher variety of scenes may increase the performance of a learning-based MVS system when it has been shown deep learning methods benefit greatly from being given more data.

Addressing the above concerns, this working paper presents a synthetic dataset of approximately 9,000 images from a variety of virtual Unreal Engine 4 environments along with the novel data collection pipeline used to collect these images. Additionally, preliminary results of a baseline deep learning model trained on this dataset are presented. Lastly, this working paper presents an in-depth discussion on a novel deep learning architecture that decomposes input images into a hierarchical structure of progressively lower resolution scales using a Laplacian Pyramid, processes each resolution scale sparsely, and then reconstructs the depth image to preserve high resolution details as well as reduce visually redundant pixels without unnecessary computation.

[1]Conner Pulling is a Robotics Institute Summer Scholar and first year Master's of Robotics (MSR) Student at Carnegie Mellon University. cpulling@andrew.cmu.edu

[2]Yaoyu Hu is a Postdoctoral Researcher with AirLabs at the Robotics Institute in Carnegie Mellon University. yaoyuh@andrew.cmu.edu

[3]Sebastian Scherer is an Associate Professor at the Robotics Institute in Carnegie Mellon University and the Principal Investigator at AirLabs. basti@andrew.cmu.edu

II. RELATED WORK

A. Binocular Stereo Vision

Recent deep-learning based dense stereo models demonstrate good accuracy when computing resources are sufficient. Binocular stereo depth estimation is widely applied for robot perception due to its simple hardware requirement and efficiency. The job of dense stereo vision is to assign a disparity value to all the pixels in the reference image. Current state-of-the-art binocular stereo model delivers superior performance in terms of accuracy compared with classical methods such as the SGM method [8]. From these learning based models we can identify some key model structures that lead to better accuracy. Among all the structures, the cost volume is the most effective way of arranging the visual features. Models such as the GCNet [9], PSMNet [10], and HSM [11] all benefit from the cost volume. While being effective, the cost volume consumes large amount of GPU memory and computing time which also imposes a major challenge for applying it to real-world robots with edge devices. Models using correlation methods [12] instead of a cost volume may have better efficiency but they are less accurate in general. To improve efficiency, many models exploit the multi-scale structure [13], [14] or cascade refinement [15], [16]. Based on the multi-scale structure, Chang, *et. al.* [17] further applies attention mechanism to boost the representational ability of the extracted visual features. Chang, *et. al.* [18] achieve real-time performance by leveraging non-learning procedures and shifting some load to special computing hardware. We are inspired by these work and we extend some of the key structures to be more efficient to meet our requirements. Notably, the above learning based methods' inference time on an edge device normally range from 80-300ms. And that was for a single stereo image pair with about 0.5M pixels in resolution.

B. Omni-directional Multiview Stereo Vision

In hopes of better aggregating information from multiple stereo pairs, some works have turned to using learning-based approaches inspired by the performance gains that deep learning has achieved for binocular stereo vision applications [10], [11]. SweepNet proposed warping the input fisheye images into the panorama space and used pairwise matching to compute the cost volume [19]. OmniMVS proposed warping the feature maps after strided convolutions to reduce memory and resource consumption before using a encoder-decoder architecture to regularize the cost volume [3]. CrownConv proposed projecting the fisheye image onto an icosahedron and using icosahedron-based spherical sweeping to be more computationally efficient [5]. However, none of the current learning-based omni-directional stereo vision methods using multiple fisheye images have achieved real-time performance. Recent work has developed a real-time non-learning method to perform omni-directional depth prediction, but does so by selectively using only the best stereo pair per pixel in a reference image and computes the cost volume only by using pixel intensities [4].

C. Synthetic Multiview Fisheye Datasets

The most notable omni-directional depth prediction datasets are OmniThings and OmniHouse [3] where each sample consists of four fisheye camera images that were generated in Blender. However, the dataset assumes fixed camera intrinsics and extrinsics, so the dataset is only useful for a specific camera configuration. This work provides the base panorama images as part of the presented dataset so that other works can warp the source into fisheye images specific to their camera orientations.

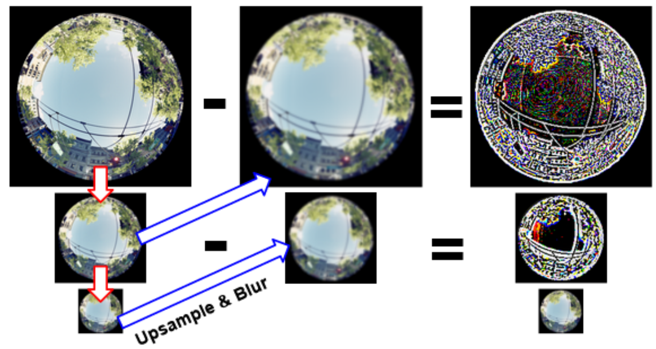


Fig. 1. **Laplacian Pyramid Process.**

D. Sparse Convolution and Image Decomposition

In many computer vision tasks, query images are quite dense and contain an abundance of visual information. This abundance of visual information can be seen in image classification and segmentation datasets such as ImageNet and COCO that contain hundreds of thousands or even millions of detailed and colorful images [20], [21]. For classification and segmentation tasks, these details are essential for a network to extract more informative features. However, for images with a low number of features, most of the image is not useful. Therefore, by only processing the parts of the image that are useful, computer vision methods using this sparse representation would be more efficient. Recent sparse convolution methods show the potential for great speedups as well as performance increases in sparse spatial learning-based tasks such as handwriting classification and point clouds [22]–[24]. One way to partially sparsify images is through thresholding the image pyramids produced by image decomposition techniques such as Laplacian Pyramids. Laplacian Pyramids are a variant of Gaussian Pyramids, where rather than simply downsampling and blurring the resultant image, images are first blurred and then subtracted against the original image before preceding down the pyramid. The resulting images, except the bottom of the pyramid, almost represent edge maps where higher magnitude pixel values indicate edges, texture, and other high-spatial frequency features. Conversely, pixel magnitudes close to zero may represent smooth or textureless areas within the image. The process of computing the Laplacian Pyramid is presented in Figure 1.

Fig. 2. **Overview of the Nine Environments Collected.** 1,000 samples were collected per environment using the novel flexible data collection pipeline.

III. DATASET GENERATION

A. Camera Configuration

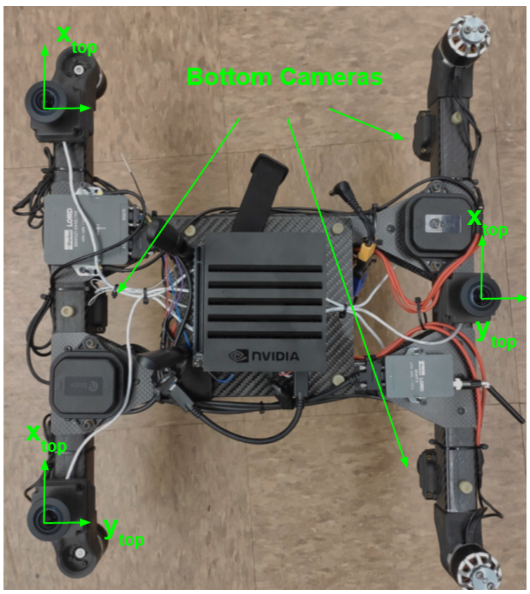


Fig. 3. **UAV Camera Configuration.** At each green coordinate frame, the top fisheye cameras are indicated. Each green arrow indicates a fisheye camera that has been mounted underneath the UAV.

In prior works, four fisheye cameras are used to sample images from the scene and are arranged in 90 degree intervals such that the front of the camera points out into the scene [3], [4]. This outward-facing camera configuration makes use of the best part of the fisheye image, which is the center. Because the fisheye is a circular image, the resolution of the image is not constant and decreases radially along with fisheye image. Thus, when the fisheye lens cameras are pointed directly at the interesting parts of the scene, depth estimation methods receive more details and edges to make better depth predictions. However, mounting cameras on UAVs is challenging. As shown in Figure 3, it is increasingly more common to mount cameras on top of the drone such that the camera is facing perpendicular with respect to the forward direction of the drone.

This design choice means that the forward direction of the drone is represented by low resolution fisheye pixels. Thus, lower resolution pixels represent most of the interesting objects in front of the UAV during level flight. With low spatial resolution, only a lower resolution depth map can be generated without using the network to create information out of thin air. At the same time, the high resolution pixels in the fisheye image are often pointed towards the ground or the sky. Thus, this UAV camera configuration is not optimal for depth estimation and adds difficulty to the problem as compared to outward-facing camera configurations in previous works.

B. Dataset Content

Our novel dataset consists of approximately 9,000 samples that have been sampled from nine Unreal Engine game environments. Using the AirSim plugin, a simulated drone is flown around the game environment and a virtual camera collects RGB panoramic images [25]. For each sample location within the environment, three images are collected to simulate the top half of the camera configuration discussed in the previous section. Because the UAV camera configuration is symmetric, a model trained the top camera arrangement approximately learns to inference on the bottom camera arrangement. In addition to the three RGB panorama images, a ground truth pixel-perfect depth image is collected. Approximately 1,000 samples are collected per environment. The dataset consists of a variety of indoor and outdoor scenes, different lighting and weather conditions, and a mix of urban and natural scenes. Additionally, some scenes contain difficult features for stereo vision such as repeating textures and objects, drastic changes in lighting, randomized orientation of the simulated drone, and thin objects. Since a UAV would be severely damaged if it ran into a thin line object such as a pole or power line, the frequent inclusion of thin line objects in the dataset can be used to evaluate a model's ability to predict their depth.

C. Preprocessing

After the panorama images are sampled from the Unreal Engine environments, they need to be processed into a specific fisheye camera. The sample process requires a calibrated camera model and camera orientation to correctly project which pixels from the panorama image must be sampled.

As shown in Figure 4, the conversion process starts by initializing a normalized coordinate grid in the shape of the fisheye image. From each pixel location, a three-dimension ray is unprojected from the camera using the Double Sphere camera model [26]. The three dimensional rays are rotated to correspond to the fisheye camera's orientation with respect to the panorama reference frame. The rays are then converted

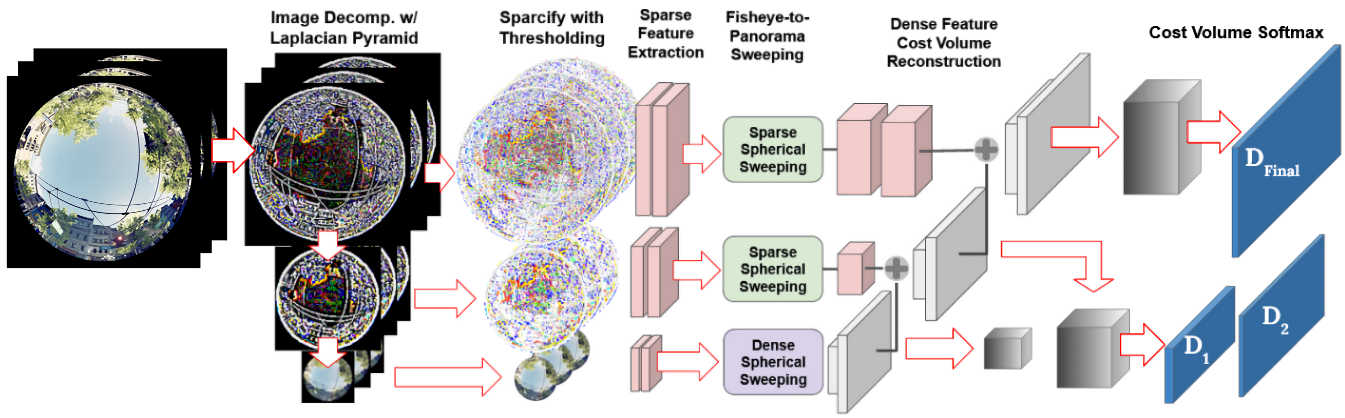


Fig. 4. **AirMVS Architecture Overview.**

to the longitude-latitude location of the ray on the panorama image. Thus, we are able to calculate the precise location of each fisheye pixel in the panorama image. When done for every pixel in the fisheye image, a sampling grid is made, which can be used to resample the panorama image into an arbitrary fisheye image.

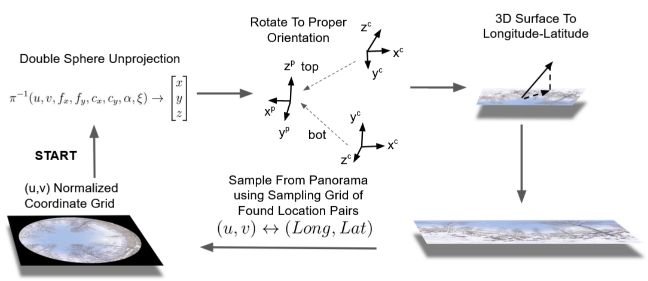


Fig. 5. **Panorama to Fisheye Image Conversion Process.**

IV. MULTI-VIEW OMNI-DIRECTIONAL STEREO

A. Architecture Overview

As shown in Figure 3, the model takes three RGB fisheye camera images and three fisheye-to-panorama sampling grids generated for spherical sweeping as input. Spherical seeping [3] is similar to the process of building a cost volume in binocular stereo models, e.g. [11]. Our model is inspired by these implementations. The input RGB images are of shape $[B, 3, H_{in}, W_{in}]$ where B is the batch size and H_{in}, W_{in} are the height and width of the inputs, respectively. The three fisheye-to-panorama sampling grids for spherical sweeping are of shape $[B, P, H_{out}, W_{out}, 2]$ where P is the number of sweepings, H_{out} and W_{out} are the height and width of the output panorama cost volume, respectively. A shared feature extractor is used by each camera with sparse convolution. Sparse Convolutional layers using residual skip connections are utilized as the basic building block in the feature extractor. During spherical sweeping, these feature maps are then warped into the rig panorama space and swept for all depth candidatesn. After the volume of the panoramas for all cameras and for all depth candidates is built, a component similar to the upsampling half of U-Net [27] is used to compute the final costs. Softargmax is used to find the final depth candidate panorama map similar to other learning based binocular stereo models.

B. Spherical Sweeping and Sparcity

As shown, the proposed architecture Spherically Sweeps the fisheye features into the panorama space to create the cost volume, first employed by OmniMVS [3]. This is a dense process that uses sampling grids generated from the camera calibration intrinsic parameters and the camera extrinsic configuration to warp a fisheye image into a panorama at specified distance candidates.

With using sparse feature vectors, a challenge is presented: what is the most efficient way to warp the sparse fisheye image into the panorama space when the model does not know which pixels need to be warped ahead of time? This paper proposes the use of a lookup table to precompute the integer location of a fisheye pixel in the panorama space. These panorama locations for each fisheye pixel at each depth candidate are stored in a three-dimensional tensor. Since the indices of each sparse feature are readily available, the lookup table simply needs to be indexed with the fisheye pixel location and depth candidate index to retrieve the corresponding panorama location. While very computationally efficient, projection errors are introduced as the floating point panorama location is truncated to an integer. Future work will look to interpolate the floating point panorama location into the corresponding integer locations by adding active sites to the sparse feature vector.

V. PRELIMINARY QUALITATIVE RESULTS

Using a model architecture similar to OmniMVS, a baseline model was trained on our novel dataset. A few of the inferenced images that represent the total performance of the network are presented in Figure 6.

VI. CONCLUSIONS

This paper has described the process of generating a multiview stereo vision dataset consisting of raw RGB and

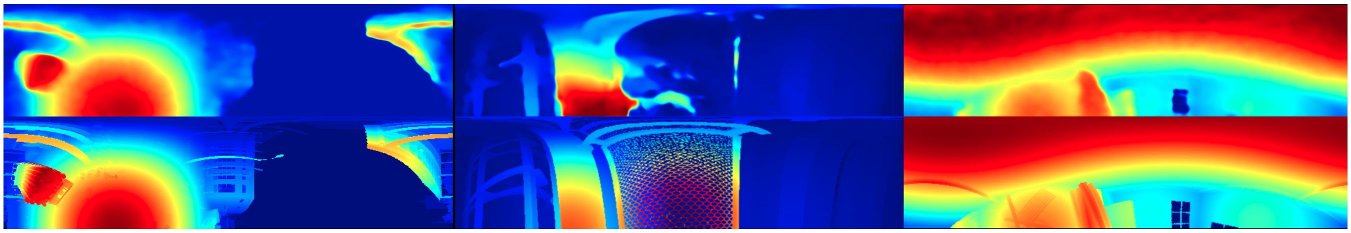


Fig. 6. **Preliminary Depth Images generated by the baseline model.** Each set of images includes the true depth image (bottom) and the predicted depth image output from the model (bottom). Warmer colors indicate closer depth. The more that the predicted depth map resembles the true depth, the more accurate the baseline model is.

ground truth depth map panoramas in Section 3. Additionally, this paper presents the contents of the synthetic dataset generated for training a novel deep learning MVS model, whose architecture has been illustrated in Section 4. Overall, this paper represents continuing work in the field of fast learning-based multiview stereo vision using fisheye images as input. Future work includes the evaluation of SOTA methods trained on the generated dataset to show that training on our novel and larger dataset can produce better accuracy as compared to SOTA datasets. Additionally, future work will implement the architecture described in Section 4 and present the qualitative and quantitative results of using the model for the MVS task described in this paper.

ACKNOWLEDGMENT

The authors would like to thank Dr. John Dolan and Rachel Burcin for coordinating the RISS program this summer and for providing the opportunity to conduct this research at Carnegie Mellon University (CMU). This research was graciously funded and supported by the AirLab at CMU. Special thanks to Wenshan Wang, Zifa Zhu, and Mohit Singh for the work on building the simulated environments. The authors also want to thank Junyi Geng, Zheng Xu, and Ruohai Ge from CMU for their valuable work on the hardware and data collection.

REFERENCES

[1] B. F. Spencer, V. Hoskere, and Y. Narazaki, "Advances in computer vision-based civil infrastructure inspection and monitoring," *Engineering*, vol. 5, no. 2, pp. 199–222, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2095809918308130

[2] M. Vázquez-Arellano, H. W. Griepentrog, D. Reiser, and D. S. Paraforos, "3-d imaging systems for agricultural applications—a review," *Sensors*, vol. 16, no. 5, 2016. [Online]. Available: https://www.mdpi.com/1424-8220/16/5/618

[3] C. Won, J. Ryu, and J. Lim, "Omnimvs: End-to-end learning for omnidirectional stereo matching," *CoRR*, vol. abs/1908.06257, 2019. [Online]. Available: http://arxiv.org/abs/1908.06257

[4] A. Meuleman, H. Jang, D. S. Jeon, and M. H. Kim, "Real-time sphere sweeping stereo from multiview fisheye images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[5] R. Komatsu, H. Fujii, Y. Tamura, A. Yamashita, and H. Asama, "360° depth estimation from multiple fisheye images with origami crown representation of icosahedron," *CoRR*, vol. abs/2007.06891, 2020. [Online]. Available: https://arxiv.org/abs/2007.06891

[6] P. K. Lai, S. Xie, J. Lang, and R. Laganière, "Real-time panoramic depth maps from omni-directional stereo images for 6 dof videos in virtual reality," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2019, pp. 405–412.

[7] J. Keller and S. Scherer, "A stereo algorithm for thin obstacles and reflective objects," 2019. [Online]. Available: https://arxiv.org/abs/1910.04874

[8] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, 2005, pp. 807–814 vol. 2.

[9] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[10] J. Chang and Y. Chen, "Pyramid stereo matching network," *CoRR*, vol. abs/1803.08669, 2018. [Online]. Available: http://arxiv.org/abs/1803.08669

[11] G. Yang, J. Manela, M. Happold, and D. Ramanan, "Hierarchical deep stereo matching on high-resolution images," *CoRR*, vol. abs/1912.06704, 2019. [Online]. Available: http://arxiv.org/abs/1912.06704

[12] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4040–4048.

[13] P. L. Dovesi, M. Poggi, L. Andraghetti, M. Martí, H. Kjellström, A. Pieropan, and S. Mattoccia, "Real-time semantic stereo matching," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 780–10 787.

[14] A. Tonioni, F. Tosi, M. Poggi, S. Mattoccia, and L. D. Stefano, "Real-time self-adaptive deep stereo," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[15] X. Jia, W. Chen, Z. Liang, X. Luo, M. Wu, Y. Tan, and L. Huang, "Multi-scale cascade disparity refinement stereo network," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 2110–2114.

[16] H. Dai, X. Zhang, Y. Zhao, H. Sun, and N. Zheng, "Adaptive disparity candidates prediction network for efficient real-time stereo matching," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[17] J.-R. Chang, P.-C. Chang, and Y.-S. Chen, "Attention-aware feature aggregation for real-time stereo matching on edge devices," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.

[18] G. Chen, H. Meng, Y. Liang, and K. Huang, "Gpu-accelerated real-time stereo estimation with binary neural network," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 12, pp. 2896–2907, 2020.

[19] C. Won, J. Ryu, and J. Lim, "Sweepnet: Wide-baseline omnidirectional depth estimation," *CoRR*, vol. abs/1902.10904, 2019. [Online]. Available: http://arxiv.org/abs/1902.10904

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[21] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014. [Online]. Available: https://arxiv.org/abs/1405.0312

[22] H. Tang, Z. Liu, X. Li, Y. Lin, and S. Han, "Torchsparse: Efficient point cloud inference engine," 2022. [Online]. Available: https://arxiv.org/abs/2204.10319

[23] B. Graham and L. van der Maaten, “Submanifold sparse convolutional networks,” 2017. [Online]. Available: https://arxiv.org/abs/1706.01307
[24] B. Graham, M. Engelcke, and L. van der Maaten, “3d semantic segmentation with submanifold sparse convolutional networks,” 2017. [Online]. Available: https://arxiv.org/abs/1711.10275
[25] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “Airsim: High-fidelity visual and physical simulation for autonomous vehicles,” *CoRR*, vol. abs/1705.05065, 2017. [Online]. Available: http://arxiv.org/abs/1705.05065
[26] V. Usenko, N. Demmel, and D. Cremers, “The double sphere camera model,” *CoRR*, vol. abs/1807.08957, 2018. [Online]. Available: http://arxiv.org/abs/1807.08957
[27] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
[28] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, “Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 4304–4311.
[29] K. Matzen, M. F. Cohen, B. Evans, J. Kopf, and R. Szeliski, “Low-cost 360 stereo photography and video capture,” *ACM Trans. Graph.*, vol. 36, no. 4, July 2017. [Online]. Available: https://doi.org/10.1145/3072959.3073645
[30] H.-S. Lin, C.-C. Chang, H.-Y. Chang, Y.-Y. Chuang, T.-L. Lin, and M. Ouhyoung, “A low-cost portable polycamera for stereoscopic 360° imaging,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 4, pp. 915–929, 2019.
[31] Y. Hu, W. Zhen, and S. A. Scherer, “Deep-learning assisted high-resolution binocular stereo depth reconstruction,” *CoRR*, vol. abs/1912.05012, 2019. [Online]. Available: http://arxiv.org/abs/1912.05012
[32] M. Menze and A. Geiger, “Object scene flow for autonomous vehicles,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3061–3070.
[33] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “Aerial Informatics and Robotics platform,” Microsoft Research, Tech. Rep. MSR-TR-2017-9, 2017.
[34] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer, “Tartanair: A dataset to push the limits of visual slam,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4909–4916.

Object Detection & Mapping for a Robotic Harvester

Ekechi Nzewi[1], George Kantor[2] and Abhisesh Silwal[3]

***Abstract*— Object detection and tracking enables yield estimation in apple orchards to be more informed. Our goal is to be able to improve on previous advancements in fruit detection and tracking. Previous research employed the use of a monocular camera to achieve performance comparable to a state-of-the-art system which uses an expensive sensor suite. Our system achieves this by employing the use of stereo cameras with bounding box detection to supplement apple tracking. Our pipeline starts with a stereo camera capturing the left and right images of its current subject. The left image is then sent to a CNN trained on data from the same orchard and bounding boxes are generated for the apples in the image. In parallel, a disparity map is generated from the left and right images. The disparity map, along with the bounding boxes, is then used to generate a depth map of the scene. The depth map was then supposed to be used for EKF-SLAM, but we have been unable to complete the pipeline. As a result, the next direction for use would be to complete the pipeline and see how our pipeline fares against others. In addition, we plan to incorporate instance segmentation to further improve the accuracy of the model.**

***Index Terms*— Computer Vision for Automation, Field Robots, Robotics and Automation in Agriculture and Forestry**

I. INTRODUCTION

Neural Networks have led to improvements in the field of computer vision. Such improvement can be seen in the field of object detection and tracking especially in terms of performance [1]. As a result of this, there has been an increase in the use of computer vision for agriculture. The core improvements have been in tasks such as yield estimation. Yield estimation works by using fruit detection and counting models to generate an estimate of the crop yield for that season. Fortunately, there exist many state-of-the-art paradigms suitable for this task [2].

Our goal is to apply these techniques to supplement a robotic harvesting system with a focus on apple harvesting. Our implementation necessitates a system capable of taking a frame from an apple orchard, detecting the apples in the frame, and mapping these detections to the reference frame of the robotic harvester. This will enable the harvester to determine the locations of apples in need of harvesting. Our pipeline incorporates object tracking and active canopy agitation to supplement harvesting. The active canopy agitation will be done through the use of a leaf blower which will agitate to ensure all apples are being detected properly and are not obscured by leaves in the frame, while the object tracking will be enabled through the use of EKF-SLAM which ensures that detected apples persist across multiple frames to prevent multiple detections of the same apple.

The structure of this paper is as follows: In section II, we explain the background pertaining to our system, showing related works, and our object detection and tracking architectures. In section III, we explain the overview of our system, our contribution to the system and the methodology of our system. Section IV contain our results and evaluations of our models. Section V will be for future work and discussions.

Fig. 1. Example of Frame being analyzed

II. BACKGROUND

A. Related works

1) General Object Detection: Research in object detection and tracking architectures has grown prevalent in recent times. Applications of object detection and tracking can be seen in medical imaging, automated robotics, image recognition and even surveillance systems. Traditional object detection works by informative region selection, feature extraction, and classification. Traditional region selection is done with a sliding window approach. This method works by taking exhaustive sliding rectangular "patches" of fixed width and height for each image. Feature extraction then happens on each derived patch. After which a classifier is used to distinguish between objects in each frame. Due to the exhaustive nature required with the sliding windows, this traditional method is ineffective with real-time analysis. Nowadays, with the prevalence and utilization of Convolution Neural Networks (CNN) and deeply trained models, detections algorithms can occur at a much faster rate. We briefly discuss these state-of-the-art models in a later section.

[1]Ekechi Nzewi is a student in the Computer Science Department at Georgia Institute of Technology, Atlanta, GA, USA enzewi3@gatech.edu

[2]George Kantor is with the Field Robotics Center, Carnegie Mellon University Robotics Institute, Pittsburgh, PA, USA gkantor@andrew.cmu.edu

[3]Abhisesh Silwal is with the Field Robotics Center, Carnegie Mellon University Robotics Institute, Pittsburgh, PA, USA asilwal@andrew.cmu.edu

2) Image Segmentation: Image segmentation is an extension of image classification where, in addition to classification, we perform localization. Image segmentation thus is a superset of image classification with the model pinpointing where a corresponding object is present by outlining the object's boundary.

3) Monocular Based Fruit Counting and Mapping: Liu et al. present a cheap, lightweight, and fast fruit counting pipeline [3]. Their pipeline relied on a monocular camera and achieved comparable performance to a state-of-the-art system with a more extensive sensor suite. Their pipeline begins with a detection component that uses convolutional neural networks (CNNs). It then tracks fruits and tree trunks across images, with a Kalman Filter fusing measurements from the CNN detectors and an optical flow estimator. Finally, fruit count and map are estimated by an efficient fruit-as-feature semantic structure from motion (SfM) algorithm which converts 2D tracks of fruits and trunks into 3D landmarks, and uses these landmarks to identify double counting scenarios.

B. Object Detection Architectures

1) YOLO (You Only Look Once): Redmon et al. [4] propose a regression approach to object detection that requires only a single look at an image for object detection. It consists of 24 convolutional layers and two fully connected layers and as the name suggests, it requires only one single forward propagation through the layers to detect objects. When compared to the architectures of RCNN, it tends to make more localization errors, but false positives are far less likely. In terms of speed of processing, YOLO's base model easily outperforms the already fast Faster RCNN–processing at 45 frames per second (fps). YOLO like RCNN comes with other versions, with its fast version processing at more than 150 fps. With such a massive processing rate, it is very suitable for our tasks wherein a robotic harvester will be moving across an apple orchard detecting apples for harvesting.

2) Mask R-CNN: He et al. [5] present a framework for object instance segmentation. It generates high-quality segmentation masks for each instance of an object. The Mask R-CNN framework does this by framework works by extending Faster R-CNN to predict an object mask while still performing bounding box recognition. When compared to Faster R-CNN, Mask R-CNN adds a small overhead running at 5 fps while being simple to train. Moreover, Mask R-CNN is easy to generalize to other tasks. Mask R-CNN showed top results in all three tracks of the COCO suite of challenges, including instance segmentation, bounding-box object detection, and person keypoint detection. Without bells and whistles, Mask R-CNN outperforms all existing, single-model entries on every task, including the COCO 2016 challenge winners.

III. SYSTEM OVERVIEW

Our system contains two major pipelines: the detection pipeline and the image pipeline. The image pipeline makes use of a stereo camera to capture a stereo image of the orchard we are working with. These stereo images are then passed through the detection pipeline which analyzes them and generates detections based on the apples found in the image. These detections are then used to map the apples to the reference frame of the robotic harvester. The detection pipeline is the continuation of our previous work. The previous system involved a more manual setup wherein the user had to hand label the frames being analyzed. But our new system generates the detections based on the models we trained. This enables us to set a threshold for the detections generated which allows us to perform tasks such as active canopy agitation. By setting a threshold for what is considered a detection, we can set the leaf-blower to send pressurized air towards detections that border the threshold. The air would hopefully clear leaves out of the way so we can get better detections with the models. Once the final detections have been made, the bounding boxes, along with the stereo images, are passed through parts of the original ROS pipeline. The images are used to generate a disparity map which is then used to generate a depth map. The pixels belonging to the bounding boxes are then extracted from the depth map, and from these, the pixels belonging to the apples are further extracted. These are then used as a new depth map for the system. The depth map information and bounding box coordinates are then used for tracking and mapping the locations of apples in frames.

A. Our Contribution

Our contribution is centered around automatically detecting the apples. Previously, one would have needed to manually label the images from the camera to achieve a result. Our system, however, takes the already labeled files and uses them as the ground truth to train a model for detecting new apples in new files. We trained two models with the intention of comparing their performance: YOLOv5 and Mask R-CNN. YOLOv5 focuses specifically on object detection, and Mask R-CNN was used for image segmentation. In the updated ROS pipeline, the image is passed through the model which detects the bounding boxes and passes that for use in generating the new depth map.

B. Methodology of Our System

1) Grab and read the left and right images from the camera.
2) Apply an object detection model to the frame for the left image.
3) Get and store bounding boxes and scores for each instance of detected apple.
4) If the confidence score is less than .6, try to get a better score by letting the leaf blower blow away the portions obstructed by leaves.
5) Update the bounding boxes and scores with the values gotten after using the leaf blower.
6) Generate a disparity map from the left and right images.
7) Generate a depth map from the disparity map.

8) Using the depth map and the bounding boxes, extract the pixels for the apples and store it as a new depth map.
9) Using the new depth map, track the apples across all frames and return the position of the apples.

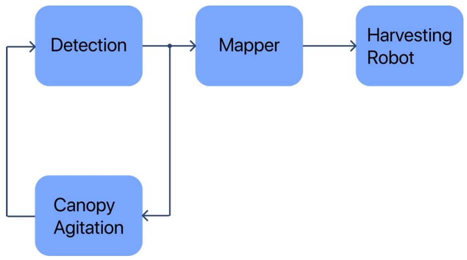


Fig. 2. System Methodology

IV. RESULTS

We tested on YOLOv5 small and YOLOv5 medium for our YOLOv5 models. On average, the mean average precision (mAP) for the models were above 0.8 with YOLOv5 performing better on average. However, we noticed that these scores may have been a result of overfitting because the model failed to generalize well on unseen data. To try and prevent overfitting, we augmented the training, and reduced the epoch size, but the model still failed to generalize to unseen data. To try and understand why it was overfitting, I looked through the data and noticed the training data was nearly identical. Thus, I devised a new solution to combat the overfitting. I randomly sampled the entire dataset to get a more diverse training set and began training on the new data. I trained a new YOLOv5s model on the new dataset and noticed that though the mAP dropped, around 0.7 now, the model then generalized better to unseen data.

This same phenomenon was noticeable with the Mask R-CNN model. Not only was the model unable to generalize to unseen data, but the model also failed to perform well on the training/testing data. The same approach as above was taken to improve the performance of the Mask R-CNN models. And though the results were marginally better, there is still room for improvement. We suspect the model performed as it did because of the occlusion of certain apples by leaves. Because of the nature of image segmentation which requires one to label the pixels belonging to the object being detected, it was difficult to accurately separate the leaf pixels from the apple pixels during training which probably affected the performance of the model. Despite all this, the Mask R-CNN model was able to predict some of the same apples that were predicted by the YOLOv5 model as can be seen in Fig. 3 and Fig. 4.

Fig. 3. Yolov5 Results

Fig. 4. Mask R-CNN Results

V. FUTURE WORK AND DISCUSSION

As was mentioned above, the Mask R-CNN model has significant room for improvement. An avenue that was not explored for increasing the performance of the model is modifying training parameters. Due to time constraints, the Mask R-CNN model was only able to be trained with the bare minimum settings. For example, only the head of the model was trained as opposed to the whole model, and the model was only trained with the mask data and not the bounding box data. By varying these settings, and others such as training time, we aim to improve the model to performance similar to that of the YOLOv5 model.

Fig. 5. YOLOv5 Ground Truth vs Predictions

Another avenue we are looking at is implementing apple tracking through EKF-SLAM. The same time constraints mentioned above led to us being unable to implement tracking across frames. As of now, the model, if deployed, will re-detect previously detected apples because there is not object

< TABLE OF CONTENTS

permanence across the frames being fed into the model. By implementing tracking, we can mitigate the performance costs that arise from continually detecting previously detected apples. This also provides a benefit for the robotics harvester because without apple tracking, errors may accrue from the robot trying to harvest apples that were there in one frame, and gone in the next or vice verse.

Finally, the implementation of the active canopy agitation is needed. Currently, we have the necessary hardware to perform the canopy agitation, but it has yet to be implemented in the pipeline. With all of these additions, all that would be left would be fine-tuning the model's performance.

ACKNOWLEDGMENT

This work was sponsored in part by the CMU Robotics Institute Summer Scholars Program and USDA/NSF NRI 2020-67021-30760. I would like to thank Dr. George Kantor and Abhisesh Silwal for mentoring and supporting me throughout all phases of this project. I would also like to thank Rachel Burcin, and John M. Dolan for their work in making this experience possible.

REFERENCES

[1] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, Arizona, USA, 2016, pp. 3464–3468.

[2] N. Häni, P. Roy, and V. Isler, "A comparative study of fruit detection and counting methods for yield mapping in apple orchards," *Journal of Field Robotics*, vol. 37, pp. 263–282, 2020.

[3] X. Liu, S. W. Chen, C. Liu, S. S. Shivakumar, J. Das, C. J. Taylor, J. Underwood, and V. Kumar, "Monocular camera based fruit counting and mapping with semantic data association," *IEEE Robotics and Automation Letters*, vol. 4, pp. 2296–2303, 2019.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015. [Online]. Available: https://arxiv.org/abs/1506.02640

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," 2017. [Online]. Available: https://arxiv.org/abs/1703.06870

Learning Vehicle Dynamics through Interactions for Off-Road Driving

Parv Maheshwari[1], Samuel Triest[2], Wenshan Wang[2] and Sebastian Scherer[2]

***Abstract*— In practical tasks such as off-road driving, robots need to understand the physical properties of the world to deal with its complexity. The aim of our work is to enhance the performance of the vehicle model of a modified Yamaha Viking ATV, by learning from its interactions with an ever-changing environment. We show that this results in increased model fidelity in challenging scenarios such as loose soil, traversing over vegetation, etc. We perform traditional offline system identification for our vehicle model combined with two approaches for online system identification - a traditional approach and a novel learning-based method. We then move on to compare the accuracy of these vehicle models on real-world data.**

***Index Terms*— Model Learning for Control, Field Robots, Autonomous Vehicle Navigation**

I. INTRODUCTION

For off-road navigation, robots often have to perform aggressive maneuvers on rough terrain. Not only this, but the vehicle needs to adapt to changing environments and terrain. Hence there is a need for a robust and adaptive vehicle model which would allow the predicted future state of the robot to be as close as possible to the ground truth. In history, such an adaptive model is achieved by using system identification.

System identification aims to find a set of parameters (P) to best describe the vehicle model on the basis of given information. To the best of the authors' knowledge, even though there are no direct works on parameter estimation of vehicle models in the off-road driving domain, the past works show promising results of system identification in various other applications. These applications are not just restricted to on-road driving [1] as [2] leverages real-world data for modeling an industrial car-like tractor. There also exists use cases of system identification in both aerial [3] and underwater vehicles [3], [4]. In literature, mainly the works on system identification and parameter estimation of vehicle models can be categorized into offline and online approaches.

We formulate the problem of traditional system identification similar to [5] which encourages the use of traditional offline approaches like using the least squares methods to estimate the value of the unknown parameters. We use a gradient-based optimizer [6] to minimize our loss.

While the offline approaches have seemed to work fairly well in the past, to identify the parameters when no prior information is provided about them, some recent works like [7], [8] explicitly show the advantage of online approaches over using the offline approaches in real-life.

[1] Department of Mathematics at Indian Institute of Technology Kharagpur

[2] Robotics Institute at Carnegie Mellon University

Fig. 1. We perform system identification for a Customized Yamaha Viking ATV while traversing through various environments.

For traditional online system identification, we use the same methodology as its offline variant. Our motivation to continue using the least squares formulation with gradient descent for online system identification similar to the offline variant comes respectively from [9] and [10].

While the possibility and application of classical approaches have been well explored for system identification in both online and offline variants, on the other hand, learning-based approaches for system identification are quite uncommon. One of them is [11] which combines images from the front camera along with vehicle dynamics to learn the coefficient of friction that is used in the vehicle model. While this approach takes into account the future surroundings, it does not leverage the history of the vehicle's trajectory in any form, which can especially help a lot in determining the terrains that the vehicle has traversed on and most likely still traversing on.

With this motivation in this paper, we present a learning-based approach for online system identification which leverages the recycling history of the trajectory that the vehicle has already followed. This is done using a novel architecture for the neural network which uses the current parameters of the vehicle model along with a trajectory history to output an individual Gaussian distribution for each parameter. We further evaluate our approaches on real-world data collected similar to [12] using our testing platform as shown in Fig. 1. The result of this experiment shows that our learning-based approach is more adaptive and robust than both the online and offline variants of the traditional system identification.

The remainder of this paper is organized as follows. In Section II, we provide background on our vehicle model along with defining our aim. In Section III, we discuss the details of various approaches to System Identification. In Section IV, we present the description of our testing platform and explain about setting an experiment to measure the robustness of our approaches. In Section V we analyze the results of our experiment. Finally, in Section VI, we give concluding remarks and mention scopes for future work.

II. PROBLEM DESCRIPTION

Our primary focus is to design a vehicle model that accurately models the vehicle's dynamics taking into account its interactions with the real-world environment. We have used the 2 Wheel drive mode on the ATV which allows us to fundamentally formulate our vehicle modeling as a Kinematic Bicycle Model(KBM) as shown in Figure 2. Here, at i^{th} timestep, the KBM state (X) consists of X_i = [p^x_i, p^y_i, θ_i, v_i, δ_i] where p^x_i, p^y_i represent the position coordinates, θ_i is the yaw of the vehicle, v_i is the vehicle's velocity and δ_i is the steering angle. The control input (U) at the i^{th} timestep is represented by U_i which consists of [$\dot{v_i}$, $\dot{\delta_i}$] representing the acceleration and steering rate respectively. If the vehicle is moving at velocity v_i and rotating around an Instantaneous Centre of Rotation (ICR) with a steering of δ_i, the general vehicle dynamics can be represented as Equation 1.

$$f(X_i, U_i) = \dot{X_i} \;=\; \begin{bmatrix} v_i * cos\theta_i \\ v_i * sin\theta_i \\ (v_i \tan\delta_i)/L \\ \dot{v_i} \\ \dot{\delta_i} \end{bmatrix} \tag{1}$$

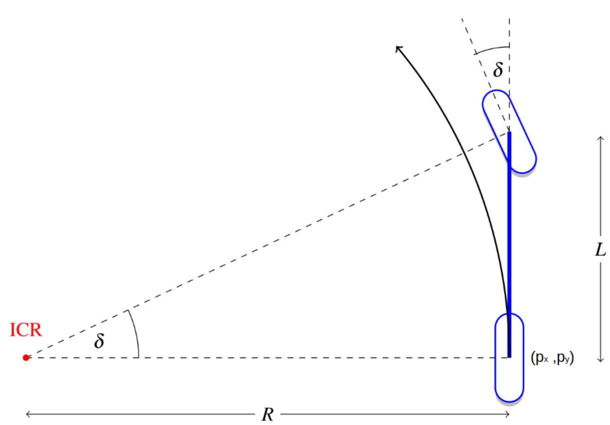


Fig. 2. Geometry of the bicycle model. The distance between the wheels is called wheelbase.

Now, at the i^{th} timestep, the next predicted state X_{i+1} can be denoted as $X_{i+1} = h(X_i, U_i)$ and can be calculated using combination of Equations 1 and 3 as f in Equation 2. For an improved estimate of the states, we use the 4th order Runge-Kutta method instead of Euler discretization.

$$\begin{aligned} h(X_i, U_i) &= X_i \;+\; \frac{1}{6}(k_1 + 2k_2 + 3k_3 + k_4) && \text{(2a)} \\ k_1 &= \Delta t * f(X_i, U_i) && \text{(2b)} \\ k_2 &= \Delta t * f(X_i + \frac{k_1}{2}, U_i) && \text{(2c)} \\ k_3 &= \Delta t * f(X_i + \frac{k_2}{2}, U_i) && \text{(2d)} \\ k_4 &= \Delta t * f(X_i + k_3, U_i) && \text{(2e)} \end{aligned}$$

where Δt is the resolution for the time step

Here we consider a modified version of the Kinematic Bicycle model where we provide throttle (T_i) and steering set point (δ_i^{target}) as actions. We define-

$$\begin{aligned} \dot{v_i} &= K_t * T_i - K_b * v_i - K_f \\ \dot{\delta_i} &= K_d * (\delta_i^{target} - \delta_i) \end{aligned} \tag{3}$$

where $P = (K_t, K_b, K_f, K_d)$ represents the set of parameters for our vehicle model. Here, K_t accounts for the effect of throttle on acceleration, $K_b * v_i$ is used to incorporate the engine braking of the vehicle as defined by [13], K_f represents the frictional force on the vehicle and K_d represents the proportional gain of the lower level steering controller. Our aim is to predict and estimate these parameters to increase the robustness and adaptive behavior of the vehicle model. These parameters can be estimated using system identification as further explained in Section III-B

III. METHODOLOGY

A. Data Collection

We have collected 30 minutes of off-road driving data in form of multiple discontinuous rosbags where each rosbag consists of multiple 5-second trajectories. This dataset aims to incorporate scenarios like acceleration, deceleration, turning, and special scenarios where the vehicle is traversing over vegetation and small rocks. These scenarios help us to find the right parameter as these cover different conditions where the effect of throttle, engine braking, and friction can influence the trajectory of the vehicle.

We have tried avoiding slopes while collecting data because as shown in Equation 3, $\dot{v_i}$ does not incorporate the effect of gravitational force, in the acceleration of the longitudinal velocity, which is non-negligible on slopes.

B. General System Identification

Here we estimate the classical vehicle model on the basis of the trajectory that the vehicle has followed. This is done by using a sequence of KBM states along with the actions as ground truth represented by $GT_{1:N}$ and $U_{1:N-1}$ respectively. Here GT_i and U_i represents the vehicle's current state and the commanded action at i^{th} timestep. Then the predicted trajectory ($S_{2:N}$) can be calculated as shown in Equation 4.

$$S_{2:N} = g(GT_1, U_{1:N-1}) \text{ where,}$$
$$g(GT_1, U_{1:N-1}) = \begin{bmatrix} h(GT_1, U_1) \\ h(h(GT_1, U_1), U_2) \\ \vdots \\ h(h(\ldots h(GT_1, U_1)\ldots), U_{N-1}) \end{bmatrix} \quad (4)$$

Given the ground truth and predicted trajectory, we calculate the loss as

$$\mathcal{L} = (GT_{2:N} - S_{2:N})^2 \quad (5)$$

We can perform system identification in two modes - offline or online.

C. Offline System Identification

We perform system identification to predict a set of values for the parameters in our vehicle model. This set can be represented as $P^o = (K_t^o, K_b^o, K_f^o, K_d^o)$ which in general can best describe the model. This is not done in real-time but rather performed on the dataset (Section III-A). For this mode, we minimize $\mathcal{L}$ to optimize the parameters set P by using the Adam optimizer [6]. The initial value of parameters can be arbitrarily set in this case.

D. Online System Identification

The online system identification similar to the offline system identification works on the history of the trajectory but unlike in the offline mode, the online mode uses real-time history to provide an updated set of parameters - $P^t = (K_t^t, K_b^t, K_f^t, K_d^t)$ at a time t. This is done because while P^o tries its best to represent the model in general, the online system identification works to provide history-specific parameters in real-time. For example, given the vehicle is traversing over pebbles and rocks, the frictional force which acts in the vehicle would be higher than what it would face while traversing over areas covered with vegetation. Hence the online system identification node would convey a higher value for K_f^t to the vehicle model than K_f^o. This helps us to increase the model accuracy in comparison to using a fixed set of parameters over different terrains and environments. The online system identification module conveys P^t to the vehicle model used by our local motion planning module as explained in Section III-F.

We have implemented Online system identification using two methods -

- Traditional Method - Similar to offline mode, we use [6] to minimize $\mathcal{L}$ for estimating P^t. Since the online system identification module has to update parameters in real-time, it is desirable for it to run at a frequency matching the frequency of the limiting observation. In our case, the limiting observation is the current position of the steering wheel which is received at 6 Hz. But due to the time taken in the forward and backward pass of a gradient-included rollout of a 5-second trajectory, in the current capacity, it is only possible to run the traditional method for a single epoch in real-time even after which this method can only run at 2 Hz.
- Learning Based Method - In this method, we use a neural network as shown in Figure 3 to predict the parameters when a history of trajectory and the current vehicle model parameters are fed into the network. Since our neural network can predict in real-time with almost little latency, we are able to use this method at a comparable rate to the limiting observation.

E. Learning-Based Online System Identification

1) Parameter Extraction: To train our architecture, we first extract out labels for P^t by using offline system identification on individual trajectories instead of using the entire collection of all the trajectories as done in the offline mode. To speed up the parameter extraction process while not hindering the accuracy of the extracted labels, we warm start the initial parameters for each trajectory with the final parameters of the last trajectory while using P^o as the initial parameters for the first trajectory for each individual rosbag.

2) Training of the architecture: As shown in Fig. 3, we represent the history of the trajectory represents the trajectory in the same KBM state space X, as explained in Section II, by processing a 5 seconds sequence of Odometry data combined with the position of the steering wheel. This KBM state history is first passed through a Wavenet encoder [14] which outputs a latent observation, which then is concatenated with the current parameters of the vehicle model. This concatenated input is passed through a multilayer perceptron (MLP) which outputs the mean and standard deviation of individual Gaussian distributions for the next set of parameters which are then selected by randomly sampling from the outputted distribution.

F. Local motion Control

We use MPPI [15] as a trajectory optimizer which provides us a local trajectory in form of a series of actions. We use our vehicle models to rollout sample trajectories in MPPI while optimizing for the following loss function -

$$\mathcal{J} = C(W_{pos}) + (\mathbb{1}_{v>=v_{v_{max}}})(e^{v-v_{max}} - 1) * (K_{penalty}) \quad (6)$$

where, W_{pos} is the position of the vehicle on the costmap, $C(p)$ is the value of the costmap at the p^{th} position, v_{max} is the maximum allowed velocity for the vehicle and $K_{penalty}$ is the speed penalty term usually kept as very high - for example - 10^8

IV. EXPERIMENTAL SETUP

A. ATV Platform

Similar to [12], various exteroceptive and Proprioceptive sensors were used to collect real-time data. We use a forward-facing Carnegie Robotics Multisense S21 stereo camera that provides us long-range high-resolution stereo RGB and depth images. For raw inertial measurements and estimates of position and velocity, a NovAtel PROPAK-V3-RT2i GNSS unit is used. As an addition to the sensors used in [12], we also use a forward-facing Velodyne LiDAR which provides us with laser scans of range up to 40m. All sensors and servos

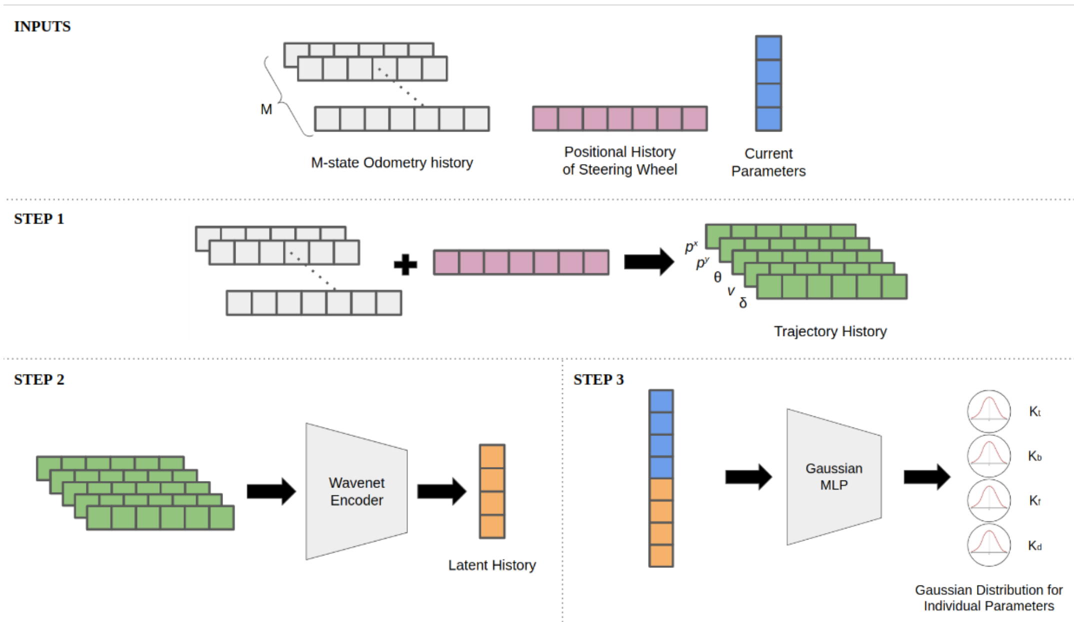


Fig. 3. Learning-based online system identification. We have used M=5.

were connected through ROS on an onboard computer. We have also relayed the joystick control inputs to the servos for the driver to manually operate the vehicle through a joystick. The sensors are integrated similar to Fig. 3 in [12].

Data from these sensors and servos are fed into our systems pipeline as shown in Fig. 4 to record the following data -

1) Robot Action: Actions $a = (\mu_t, \mu_s)$ were two-dimensional and corresponded to desired throttle and steering positions. Throttle commands took values between 0 and 1, with 1 corresponding to wide open throttle. Steering commands took values between-1 and 1, with -1 corresponding to a hard left turn. The commands were executed by the servos using PID position control.

2) Robot Pose: As an improvement to [12], instead of just using the raw measurements given by GNSS, we instead the raw measurements along with the laser scans, to run Super Odometry [16] which helps us achieve a more robust state estimation than the raw measurements from GNSS. We express the robot pose in the form of a concatenated position vector $p = (x, y, z)$, quaternion orientation $q = (q_x, q_y, q_z, q_w)$, linear velocity $v = (v_x, v_y, v_z)$ and angular velocity $w = (w_x, w_y, w_z)$. This is an improvement to [12] as we also consider linear and angular velocity and not just the position and the orientation vectors

3) Images: At each timestep, two RGB images were recorded from the onboard stereo camera.

4) Local Terrain Maps: : Similar to [12], we generate a local top-down view height map $M_h \in \mathbb{R}(w \times h \times 2)$ (two channels to represent the minimum height and maximum height) and a local RGB map $M_c \in \mathbb{R}(w \times h \times 3)$ using the stereo images from the Multisense S21 sensor and using the Stereo and Lidar Mapping Nodes. The cost maps generated from applying a lethal height threshold over the heightmaps maps are then used as explained in Section III-F.

B. Vehicle Model Accuracy

The vehicle models have been evaluated for their model accuracy on the data collected for system identification as mentioned in Section III-C. The performance has been measured in terms of the mean errors in all the individual elements in the KBM state and all of them combined. The results of this experiment have been reported in Table I.

V. RESULTS

As explained in Section III-D, to run the traditional online system identification in real time - we are only able to perform a single epoch of optimization over the previous labels. As expected this leads to a disturbance in loss $\mathcal{L}$ but since the offline optimization had already reached local minima to generally express the vehicle model, a single epoch results mostly in a downgrade of the performance rather than an improvement. This can be seen from Table I.

We further noticed that running many more optimization epochs (30 - 50) over an individual trajectory results in a P^t which is more accurate than P^o for that particular trajectory. With this motivation, we trained a neural network architecture and expected the performance, in general, of the vehicle model with Learning-based Online System Identification to be better than the other two models. This hypothesis is also confirmed from Table I.

VI. Conclusion and Future Work

We present various methods of system identification along with a novel neural network to overcome the low-frequency output of the traditional online methods. We have also verified the accuracy of these models on real-world off-road data. Moving on, along with using the existing dataset,

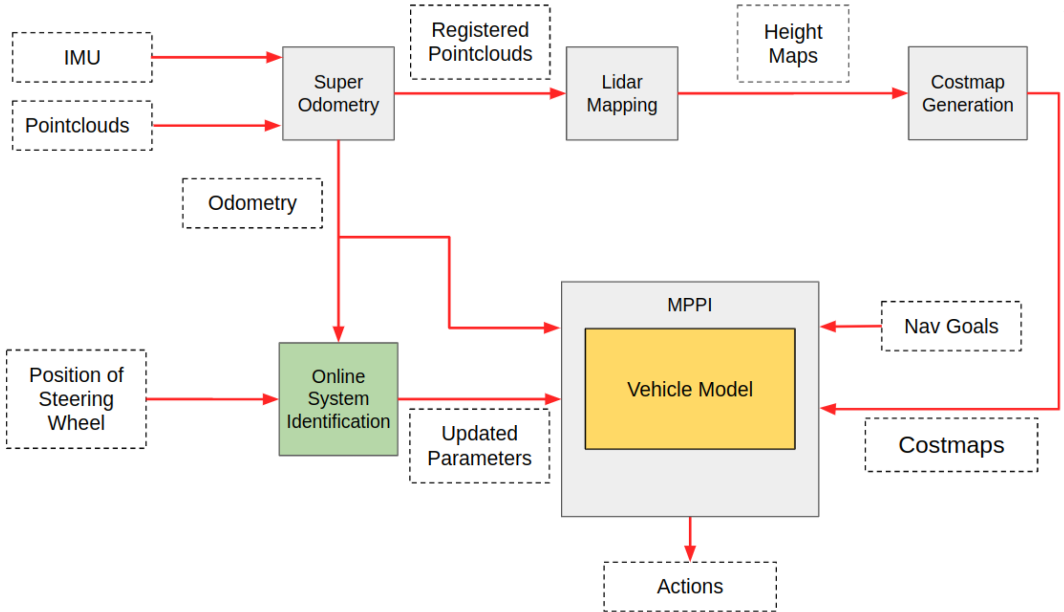


Fig. 4. Our complete Navigation stack. Here the dotted lines are only valid if the End-to-End learning-based vehicle model is used. **Remove if not explaining learning based model**

TABLE I
MEAN LOSSES FOR VARIOUS VEHICLE MODELS FOR INDIVIDUAL ELEMENTS IN THE STATE X AND ALL COMBINED

Model Type	L_{all}	L_x	L_y	L_θ	L_v	L_δ
KBM without Online System Identification	0.4285	1.4304	**0.1251**	0.0074	**0.5790**	0.0007
KBM with Traditional Online System Identification	0.5004	1.7164	0.1342	0.0075	0.6426	0.0011
KBM with Learning-based Online System Identification	**0.3189**	**0.8585**	0.1389	**0.0073**	0.5896	**0.0002**

we would also be using the entire TartanDrive Dataset [12] which is a dataset containing more than 5 hours of off-road driving data. This would not only help us achieve a more robust estimation of both the offline system identification and learning-based online system identification.

We are also motivated to incorporate additional input modalities like forward-facing terrain maps in our learning-based online system identification to also use the terrain features in the prediction of the model parameters. This would also shift the paradigm of the current online system to a more predictive-reactive approach rather than only reactive as it would use a map of the surroundings it has to traverse in the future while also using a history of states. Along with this future works can also incorporate the effect of gravity in our vehicle models. This would help our vehicle model to be more robust to changes in the pitch of the vehicle while it is traversing slopes.

ACKNOWLEDGMENT

This work was supported by the Robotics Institute Summer Scholars program at Carnegie Mellon University. The authors would like to thank Rachel Burcin and Dr. John Dolan for their support and organization of this program.

REFERENCES

[1] A. Ritter, F. Widmer, B. Vetterli, and C. H. Onder, "Optimization-based online estimation of vehicle mass and road grade: Theoretical analysis and experimental validation," *Mechatronics*, 2021.
[2] H. Zhao, W. Chen, S. Zhou, and Y.-H. Liu, "Parameter estimation of an industrial car-like tractor," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4480–4487, 2021.
[3] A. Thondiyath, "Design, estimation of model parameters, and dynamical study of a hybrid aerial-underwater robot: Acutus," pp. 423 – 430, 2019.
[4] A. Quraishi and A. Martinoli, "Online kinematic and dynamic parameter estimation for autonomous surface and underwater vehicles," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4374–4381.
[5] M. A. R. Pratama and A. Subiantoro, "Speed and yaw rate response optimization based on parameter estimation for electrical bus mathematical model," in *2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)*, 2021, pp. 1–6.
[6] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.
[7] C. Böhm, C. Brommer, A. Hardt-Stremayr, and S. Weiss, "Combined system identification and state estimation for a quadrotor uav," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 585–591.
[8] M. Cooper, J. McGree, T. L. Molloy, and J. J. Ford, "Bayesian experimental design with application to dynamical vehicle models," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1844–1851, 2021.
[9] X. Yang, C. Wu, Y. He, X.-Y. Lu, and T. Chen, "A dynamic rollover prediction index of heavy-duty vehicles with a real-time parameter

estimation algorithm using nlms method," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 2734–2748, 2022.

[10] W. Chen, D. Tan, and L. Zhao, "Vehicle sideslip angle and road friction estimation using online gradient descent algorithm," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 11 475–11 485, 2018.

[11] D. Jin, B. Leng, X. Yang, L. Xiong, and Z. Yu, "Road friction estimation method based on fusion of machine vision and vehicle dynamics," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1771–1776.

[12] S. Triest, M. Sivaprakasam, S. J. Wang, W. Wang, A. M. Johnson, and S. A. Scherer, "Tartandrive: A large-scale dataset for learning off-road dynamics models," *ArXiv*, vol. abs/2205.01791, 2022.

[13] J. Mai, "System design, modelling, and control for an off-road autonomous ground vehicle," Master's thesis, Carnegie Mellon University, Pittsburgh, PA, August 2020.

[14] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *SSW*, 2016.

[15] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control using covariance variable importance sampling," *ArXiv*, vol. abs/1509.01149, 2015.

[16] S. Zhao, H. Zhang, P. Wang, L. Nogueira, and S. Scherer, "Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 8729–8736.

Exploring Predictive Capabilities of Withdraw-Dwell-Reinfuse Cycles in Trauma Care; The Case of Blood Loss Prediction

Ernest Pokropek[1], Xinyu Li[2], Nicholas Gisolfi[2] and Artur Dubrawski[2]

***Abstract*— Trauma care is one of the most challenging tasks in modern medicine, as with no prior information about the patient's health it is extremely difficult to interpret their, notably heterogeneous, health status. Furthermore, in critical emergencies we often do not know the whole picture of the particular organism and we do not have time nor necessary equipment to perform surgeries that would enable us to collect invasive, but reliable vital sign measurements. In this work, we explore the predictive and diagnostic capabilities of Withdraw-Dwell-Reinfuse (WDR) cycles on animal data. The WDR cycle consists of, firstly, withdrawing the blood from the subject (W), then waiting (D), and finally reinjecting (R), all while collecting the vital sign measurements, for total of 60 seconds. To prove the capabilities of WDR cycles as a diagnosis tool, we present a regression pipeline that estimates the amount of blood loss given only non-invasive vital sign measurements from a single WDR cycle without prior information about the subject as we train and validate it on animal laboratory data.**

Index Terms— Medical Robots and Systems

I. INTRODUCTION

Proper estimation of blood loss might be crucial for patient's survival, as untreated hemorrhage might be fatal [1], annually causing 2 million deaths worldwide [2]. After losing approximately 20% of blood in human body, the heart is unable to pump enough blood through it, resulting in hypovolemic shock and killing the individual unless adequately treated. Proper prediction of the severity of bleeding is crucial in trauma care - with no prior access to patient's medical record, it is difficult to estimate how their bodily functions behave differently from their heterogeneous baseline. This becomes further complicated when the bleeding is internal, as we are not aware of any clues that might lead us to the correct diagnosis. Lastly, during the scenario of a traffic accident or a battlefield, it is infeasible to promptly perform a surgery to be able to collect reliable invasive measurements of vital sign data, thus requiring us to ideally utilize the non-invasive counterparts. Ideally, we would like to have a system that we can quickly "plug" into the patient and in short time retrieve information about the amount of blood loss. In order to retrieve essential from the body exposed to this case of trauma, we will treat it as a dynamical system, where a sudden intervention to it should yield essential information about its state, resilience, and condition.

[1]Ernest Pokropek is with Division of Robotics, Perception and Learning, KTH Royal Institute of Technology, Stockholm, Sweden pokropek@kth.se

[2]Xinyu Li, Nicholas Gisolfi and Artur Dubrawski are with Auton Lab, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA xinyul2@andrew.cmu.edu , {ngisolfi, awd}@cs.cmu.edu

An example of such intervention for biological system is the Withdraw-Dwell-Reinfuse (WDR) cycle. Given patient with unknown medical history, while we collect the vital sign data we perform the intervention - first, the blood is being withdrawn at a constant rate (W) for 20 seconds, ideally at a faster pace than the actual bleeding is happening. Then, for another 20 seconds, the dwell process is happening, as we hold the collected blood and wait (D). Finally, we re-inject it with the same rate as during the withdrawal, again for 20 seconds, during the reinfuse cycle (R). This intervention presents an exciting opportunity for diagnosis of the dynamical system (in our case, human body) - we expect a healthy individual to be more resilient during such stress test than a subject exposed to notable amount of blood loss, which should allow for proper prognosis using only non-invasive measurements of the vital signs. Furthermore, short duration of the WDR cycle makes it feasible to use in extremely dynamic and active environments, such as battlefield, where prompt diagnosis is crucial for survival.

In this work we address the problem of patient diagnosis with unknown medical history in the dynamic environments by utilizing vital sign data collected during bleeding of laboratory animals (pigs). We use the data from WDR cycles to train a regression model based pipeline that based on less than 2 minutes of measurements is able to assess the amount of bleeding the subject has been exposed to.

II. RELATED WORK

There is no surprise that given the importance of blood loss prediction much academic effort has been put into it. In [3], using data collected from rats, authors set up a support vector machine pipeline in order to predict the percentage volume of blood loss and calculated metrics in discretized fashion suggested by Advanced Trauma Life Support (ATLS) (severity classes based on the amount of blood loss) [4]. They reported notable accuracy of the predictions (almost 90%), however it must be noted that the study utilized invasive measurements of vital sign data (e.g. mean arterial pressure (MAP)). Another work by the same authors presented in [5] shows another approach for ATLS hemorrhage class prediction, but basing on perfusion index and lactate concentration instead of vital signs, showing once again good classification metrics with accuracy exceeding 80%.

Another, much earlier than aforementioned approach that utilized machine learning for blood loss prediction has been presented in [6]. Here, measurements of both vital signs and additional information such as hemoglobin concentration were fed into an Genetic Algorithm Neural Network

(GANN). The authors have showed 2.63% of RMS error for estimated blood volume on rat data.

Apart from machine learning based methods, there is unsurprisingly a wide variety of analytical methods for blood loss estimation, which becomes extremely useful in perioperative context. Examples of such are Moore's [7] and Nadler's [8] formulas, which take into account weight, build, and sex of the patient for estimation of the total blood volume loss. When it comes to blood loss estimation itself, one of the most common methods are involving the mean value of hematocrit (proprtion of red cells in blood), e.g. [9]–[12], often involving Moore's or Nadler's formulas.

III. METHODOLOGY

The dataset contains of 28 (2 subjects discarded due to issues with annotations) healthy pigs which were sedated for the time of the experiment. After approximately 30 minutes of stabilization with all the measurement devices connected, they were exposed to constant rate of bleeding of 10ml/min; depending on the subject, this resulted in total bleeding amount up to 1500ml. Data from 254 WDR cycles has been extracted (60 seconds of WDR + 10 seconds of offset before and 20 seconds after the WDR), of which we have used the non-invasive ECG and Pulse Oximeter Pleth measurements.

To featurize the vital sign signals, we have used windows of 10 seconds (with step size of 2 seconds), calculating the aggregated statistics from the ECG waveform, such as heart rate, approximate entropy, or powerbands obtained from Discrete Fourier Transform (DFT), whereas the photoplethysmography data has been used to derive the beat-to-beat features, such as Systolic Amplitudes, Peak-to-Peak Interval, or Pulse Interval. Furthermore, from the beat-to-beat measurements we also estimated invasive measurements, that is Mean Arterial Pressure (MAP), Stroke Volume Variation (SVV), Pulse Pressure Variation (PPV), and Dynamic Arterial Elastance. This resulted in 42 features per window. Each WDR cycle has been passed through the windowed featurization, resulting in 45 vectors of features with associated amount of bleeding (calculated at the end of the moving window).

To train and validate our pipeline we have used leave-one-subject-out cross validation methodology. For each subject, we train the Gradient Boosting Regressor model on data from 27 pigs, and validate on 1. Then, the prediction metrics are aggregated and averaged. No standardization has been used, as to try to build the model for "plug-and-play" applications in trauma care, without need of collecting baseline. For testing, we considered only 1 value of bleeding per WDR cycle in the following manner: the model predicts estimated value of blood loss for each window (10 seconds), and then aggregates all the predictions (using mean) to produce a final one. This value is then compared with the actual value of bleeding. The whole pipeline is visualized on Figure 1.

IV. RESULTS

Cumulative results from the cross validation are presented on Figure 2. We can see that the pipeline does pick the linear trend of the bleeding, although with notable deviation of the predictions. Interestingly, a consultation with medical professionals responsible for the experiments pointed out that the intercept being off (approximately 200ml) may be caused by circulation of the fluids to keep the animal alive during the experiment, which was of similar volume.

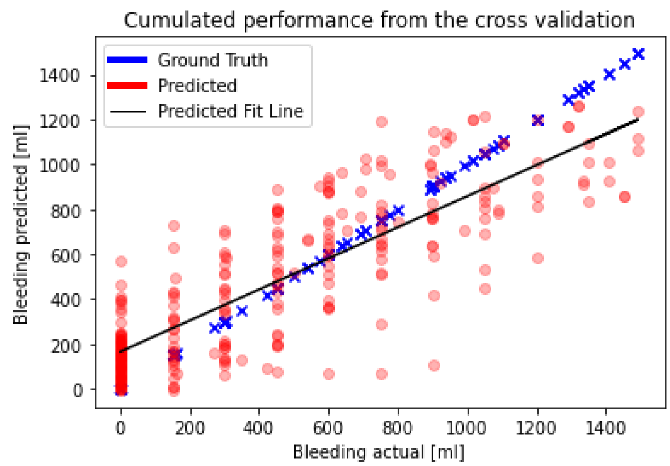


Fig. 2. Cumulated performance from all the folds of cross validation.

Another evaluation metric concerns the Advanced Trauma Life Support (ATLS) hemorrhage classes in order to classify and standardize the assessment of the patients exposed to trauma injury. Here, four classes of hypovolaemic shock are presented, with the biggest decisive factor being the fraction of the total blood lost. This is a common classification metric for blood loss prediction, and we decided to put our regression model through that evaluation as well - based on the actual value of bleeding predicted from our pipeline, the blood volume was estimated by weight of the animals and then the fraction of blood loss was calculated. The confusion matrix of this procedure is presented on Figure 3.

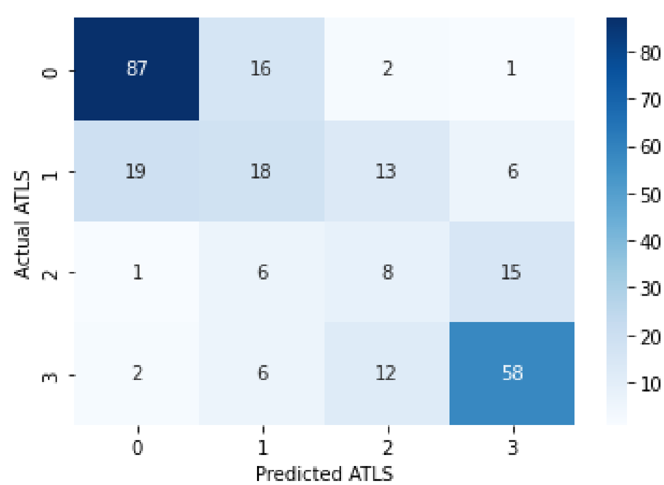


Fig. 3. Confusion matrix based blood loss predictions grouped into ATLS hemorrhage classes. The total blood volume of the animals is estimated as 1800ml.

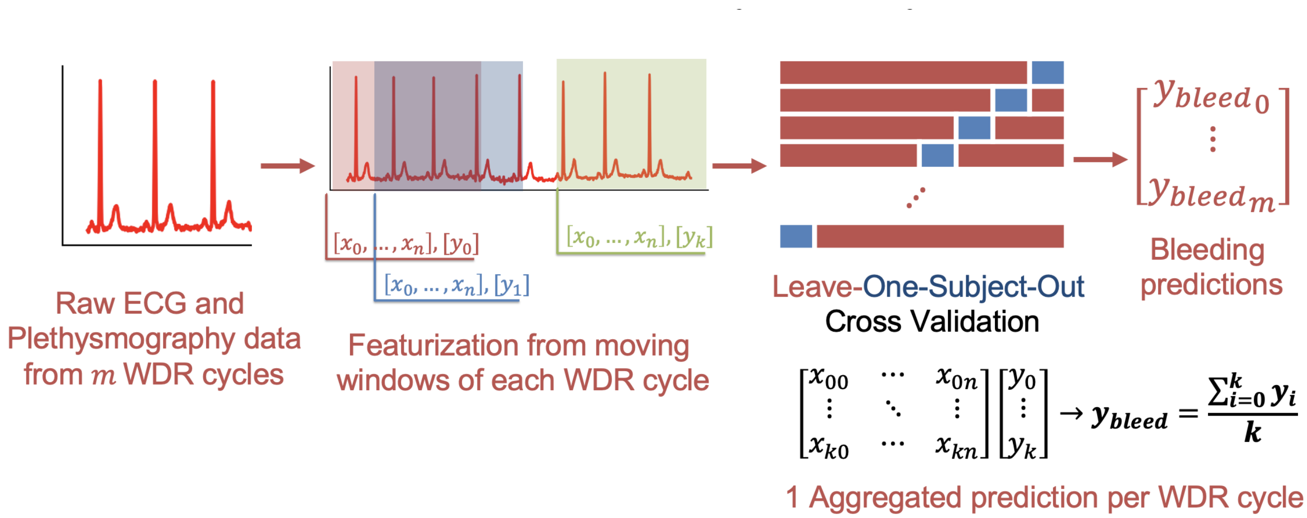


Fig. 1. Visualized pipeline of the experiment

The pipieline performs well for the extreme cases, that is first and last ATLS class - it has to be noted that they are also the most represented classes in the data. The middle classes however are not predicted so well - the possible improvements and possible drawbacks are presented in Section V.

V. DISCUSSION

In this work we presented a machine learning based pipeline that uses only quick, non-invasive vital sign measurements and is able to predict the amount of blood loss of the subject. As much as it can pick up the trend of the bleeding correctly, there are notable differences in performances between the subjects. This means, that during the leave-one-subject-out cross validation, on some test subjects the model performs incredibly well, whereas on others not as much - to illustrate that, please refer to Figure 4.

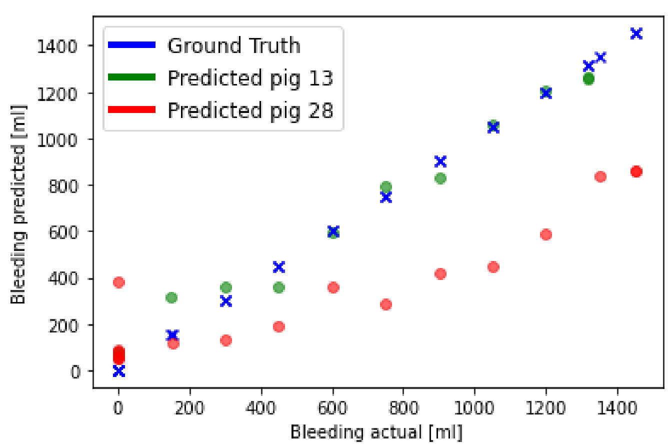


Fig. 4. Example performance on 2 different subjects showing notable differences in pipeline's performance for various subjects.

In the above figure, testing on subject 13 results in an incredibly good fit of the model, with the error of prediction within tens of milliliters. As perfect as it is, unfortunately the same trend does not follow for all the subjects - in the illustrated example, we can see that for the subject 28, the model tends to undershoot quite heavily. We suspect, that these changes in performance are caused by individual differences between the subjects, as given the trauma care application we did not use the vital sign data before the experiment to standardize it, which is a standard practise when encountering this kind of problems. Nonetheless, the pipeline performs incredibly well on some of the subjects, which gives us a lot of optimism for future works. The ideas to overcome the problem of heterogeneity are discussed in the next section, as time constraints made it impossible for us to address them properly.

VI. FUTURE WORK

This work has much potential for the future. First of all, we would like to conduct experiments that underline the potential of the WDR cycles as an intervention test: we want to compare how well the data obtained from 60 seconds of WDR cycle impacts the model predictive capabilities versus using 60 seconds outside of them. Furthermore, it would be interesting to check, how these two approaches mentioned previously work when the data is standardized to each of the animal's baseline. Also, the relationship between each cycle during the WDR needs to be explored more thoroughly - that is comparing the data from the withdraw part with dwell, dwell with reinfuse, etc. Finding some hidden relationships/correlations within the WDR cycle can immensely help the model not only with predictive capabilities, but also making it more generalizable across the individuals.

ACKNOWLEDGMENT

The first author would like to thank Xinyu Li , Nicholas Gisolfi and Artur Dubrawski for their mentorship throughout this work. Furthermore, the first author would also thank Xinyu Li for explaining the data and providing code that handles it. Finally, the first author would like to thank Rachel Burcin and John M. Dolan for perfectly managing the RISS summer internship program that made this work possible. This work was supported by by the U.S. Department of Defense under awards W81XWH-19-C-0083 and W81XWH-19-C-0101.

REFERENCES

[1] A. B. Johnson and B. Burns, "Hemorrhage," in *StatPearls*. StatPearls Publishing. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK542273/

[2] J. W. Cannon, "Hemorrhagic shock," vol. 378, no. 4, pp. 370–379, publisher: Massachusetts Medical Society _eprint: https://doi.org/10.1056/NEJMra1705649. [Online]. Available: https://doi.org/10.1056/NEJMra1705649

[3] S. B. Choi, J. Y. Choi, J. S. Park, and D. W. Kim, "Atls hypovolemic shock classification by prediction of blood loss in rats using regression models," *SHOCK*, vol. 46, p. 92–98, 2016.

[4] M. Mutschler, T. Paffrath, C. Wölfl, C. Probst, U. Nienaber, I. B. Schipper, B. Bouillon, and M. Maegele, "The ATLS® classification of hypovolaemic shock: A well established teaching tool on the edge?" *Injury*, vol. 45, pp. S35–S38, Oct. 2014, publisher: Elsevier. [Online]. Available: https://www.injuryjournal.com/article/S0020-1383(14)00375-1/fulltext

[5] S. B. Choi, J. S. Park, J. W. Chung, S. W. Kim, and D. W. Kim, "Prediction of atls hypovolemic shock class in rats using the perfusion index and lactate concentration," *Shock*, vol. 43, p. 361–368, 2015.

[6] M. F. Jefferson, N. Pendleton, S. Mohamed, E. Kirkman, R. A. Little, S. B. Lucas, and M. A. Horan, "Prediction of hemorrhagic blood loss with a genetic algorithm neural network." *Journal of applied physiology*, vol. 84 1, pp. 357–61, 1998.

[7] "Metabolic care of the surgical patient. By Francis D. Moore, M.D., Moseley Professor of Surgery, Harvard Medical School. 10¾ × 7¾ in. Pp. 1011 + x, with 14 2illustrations. 1959. Philadelphia and London: W. B. Saunders Co. 140s," *British Journal of Surgery*, vol. 47, no. 204, pp. 455–455, 12 2005. [Online]. Available: https://doi.org/10.1002/bjs.18004720451

[8] S. B. Nadler, J. Hidalgo, and T. Bloch, "Prediction of blood volume in normal human adults." *Surgery*, vol. 51 2, pp. 224–32, 1962.

[9] D. L. Bourke and T. C. Smith, "Estimating allowable hemodilution." *Anesthesiology*, vol. 41 6, pp. 609–12, 1974.

[10] C. Ward, E. Meathe, J. Benumof, and F. Trousdale, "A COMPUTER NOMOGRAM FOR BLOOD LOSS REPLACEMENT," *Anesthesiology*, vol. 53, no. 3 Suppl, pp. S126–S126, 09 1980. [Online]. Available: https://doi.org/10.1097/00000542-198009001-00126

[11] F. Mercuriali and G. Inghilleri, "Proposal of an algorithm to help the choice of the best transfusion strategy." *Current medical research and opinion*, vol. 13 8, pp. 465–78, 1996.

[12] A. Meunier, A. Petersson, L. Good, and G. Berlin, "Validation of a haemoglobin dilution method for estimation of blood loss," *Vox Sanguinis*, vol. 95, 2008.

Mixed Reality Synthetic Data Generation

Grace Su[1], Khiem Vuong[2], N. Dinesh Reddy[2], Srinivasa Narasimhan[2]

***Abstract*— Synthetic data generation augments existing vision datasets and consequently helps train more robust computer vision models. However, synthetic image generation techniques proposed by prior works still face limitations in generating photorealistic data, maintaining low computation costs, and granting fine control over scene generation parameters. In particular, synthetic data generation would be especially useful for training deep learning models for traffic analysis tasks. Therefore, we propose a photorealistic synthetic road scene generation method that inserts rendered 3D objects into a real 2D photo. We first estimate the ground plane equation, camera parameters, possible vehicle trajectories, and environment illumination map from the road scene photo. Then, these scene parameters are used to render the 3D objects in a physically-based renderer. Finally, we compose the rendered object smoothly into the road scene. Simultaneously, the renderer can generate precise depth maps. Our "mixed reality" approach's results are higher resolution and more photorealistic compared to similar previous works while addressing their limitations. Thus, our approach can generate high quality synthetic images and ground truth labels for a variety of computer vision tasks. In future work, we plan to evaluate whether our synthetic data and ground truth labels can improve deep neural network performance on challenging tasks like amodal segmentation. Code for the road scene generation method is available at `https://github.com/graceduansu/mixed_reality_synthetic_data_generation`.**

***Index Terms*— Deep Learning, Visual Perception, Object Detection, Segmentation and Categorization, Computer Vision for Automation, Synthetic Data Generation**

I. INTRODUCTION

One of the most ubiquitous challenges in developing computer vision models is obtaining realistic, accurately-labeled, diverse, and large computer vision datasets. Firstly, it is difficult, time-consuming, and expensive to acquire and label real-world data. Secondly, real-world image data is often characterized by a long tail distribution, where only a minority of different scenarios comprise the majority of collected images. This means the dataset may not encompass the full range of possible nuances and variations in each image. Synthetic data generation addresses these issues by automatically computing new images that imitate the data distributions found in existing images of the real world. The process also allows users to configure and quickly generate more diverse scenarios that are difficult to obtain from the real world while obtaining high-accuracy ground truth labels. Then, the synthetic images can augment existing datasets and consequently train more robust computer vision models.

[1] Grace Su is with the Computer Science Department, Columbia University, New York, NY., USA. `g.su@columbia.edu`

[2] Khiem Vuong, N. Dinesh Reddy, and Srinivasa Narasimhan are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA., USA. `{kvuong,dnarapur,srinivas}@andrew.cmu.edu`

For instance, creating object segmentation datasets often requires a time-consuming process where human annotators must select the regions of pixels for each segmentation mask by hand. However, if a synthetic image is generated by a computer, the ground-truth, pixel-accurate segmentation masks for each object can also be easily accessed during the generation process. Thus, a method that generates image data that is highly faithful to real image data could be useful for improving the training of many different computer vision tasks.

Additionally, when curating a computer vision dataset, we cannot rely on fully real image data because the images must be labeled by humans and humans cannot consistently produce pixel-accurate annotations. On the other hand, we cannot use fully synthetic images because machine learning models trained on such images will encounter significant domain adaptation problems (the sim-to-real gap) when they are tested on real-world images. Therefore, a "mixed reality" dataset that combines elements of real-world and synthetic image data would balance the advantages and drawbacks of both sources.

In the current literature, a number of works have generated synthetic object segmentation datasets by inserting 3D rendered objects into real-world scenes, but few attempt to take advantage of the 3D ground-truth information to generate labels for other vision tasks. In particular, synthetic data and 3D ground truth label generation would be especially valuable for vision tasks that predict 3D world information from 2D image data. One such application area is training deep learning models for traffic analysis tasks since it is important to obtain accurate data annotations and predict rare traffic patterns.

In this paper, we begin to investigate whether a mixed reality method can generate synthetic, realistic road scenes and facilitate the training of computer vision models. We chose to focus on road scene generation in order to assist training of traffic analysis-related computer vision tasks like object segmentation, 3D pose estimation, object tracking, anomaly detection, etc. Section II gives background information and reviews previous work related to synthetic image data generation, including 2D image composition, neural rendering techniques, and 3D object insertion. Section III discusses the proposed 3D object insertion and composition automated pipeline. Section IV presents and evaluates our object insertion results. Section V concludes the paper, describes potential applications, and outlines future work.

II. Related Work

A. 2D Image Composition

2D image composition methods typically "cut and paste" new objects onto desired background images, then blend the new objects into the background to make the resulting composite image look more realistic. Because this image generation method cuts and pastes images of real objects, it is closer to using fully real image data. This technique is also relatively simple and thus easier to scale. But as [1] describes, the resulting composite images are frequently unrealistic because of "appearance inconsistency (e.g., incompatible illumination), geometry inconsistency (e.g., unreasonable size), and semantic inconsistency (e.g., mismatched semantic context)." Many works address these issues by training neural networks, especially GANs (generative adversarial networks), to adjust for these inconsistencies. For instance, [2] enables object and texture editing by training a GAN to replace and blend objects. [3] also uses GAN-based models to remove and insert objects and their shadows. However, neural networks trained to blend "pasted" objects still struggle to produce consistent, photorealistic results. The networks do not always learn and apply all physical rules of photos like how perspective affects perceived object size, how occluding objects cast shadows, etc. Additionally, 2D image composition methods have no way to access 3D ground-truth information or generate such data annotations.

B. Neural Rendering Techniques

Neural rendering techniques train neural networks to learn a scene's neural radiance field (NeRF) representation and therefore produce novel views of the scene. These approaches can achieve photorealistic results by representing scenes using implicit fields of volume density and view-dependent color. Many neural rendering techniques also encode the entire scene as a whole. To ensure that the NeRF representation allows for object-level editing, [4] designs an architecture that encodes individual object information. [5] also learns object-level representations by proposing a neural rendering approach that observes a scene video, then decomposes the scene into scene graphs. These works show that NeRFs can be extended to learn object-level representations and enable object manipulation. However, there still exist several cons of using NeRFs for novel view synthesis:

- Low-interpretability and editability for dataset parameter control
- An inherent lack of 3D ground-truth information
- Higher computation costs when learning the NeRF

C. 3D Object Insertion

To generate a high-fidelity image dataset, it is advantageous to render and insert 3D, physically accurate models of objects into existing real-world backgrounds. Then, the rendering process computes the correct scene geometry, reflections, based on user-selected scene parameters, and outputs realistic images. Thus, 3D object insertion is a mixed reality approach. For example, [6] uses Blender's Cycles renderer and post-processing workflow to photorealistically render and insert 3D cars. Their results suggest that object detection and instance segmentation models trained on augmented imagery generalize better than those only trained on synthetic data or those trained on limited amounts of annotated real data. Realism of the background image also significantly affects performance. However, while their pipeline estimates the road plane and camera pose for each background image, they only use background images captured by driving cars. This means that they do not use other viewpoints like those from traffic cameras. They use a fixed set of 3D car models, locations, and environment maps to augment real street scene datasets. Their synthetic dataset and code are also not released. In addition, they only evaluate their dataset on 2D tasks (segmentation and detection) and lack benchmarks for 3D vision tasks.

Recently, Chen et al. [7] further leverages available real world data by inserting objects that were viewed with similar viewpoints and distance to the camera in its original footage. Then, they reconstruct observed objects as 3D assets and warp them to the novel target view using a differentiable neural renderer. Finally, they train a generative image inpainting synthesis network to do post-composition. When using their method for data augmentation, there are small improvements on semantic segementation performance. However, their augmentation method only uses a single cloudy environment map and does not perform lighting estimation. Their synthetic dataset and code are also not released.

On the other hand, inverse rendering and lighting estimation allows one to obtain a desired background image's scene appearance parameters that can then be used to render inserted objects with the correct lighting and geometry. For example, [8] uses a deep neural network to achieve single-image inverse rendering of indoor scenes. The network simultaneously estimates the scene's depths, normals, spatially-varying albedo, roughness and lighting, thus enabling photorealistic material editing, object insertion, and rendering. However, the estimations for depths, normals, albedo, etc. are not as accurate for outdoor scenes because there is a lack of ground-truth lighting data for outdoor viewpoints.

Overall, current works in 3D object insertion for data augmentation demonstrate small improvements for traditional object segmentation and detection tasks, but do not evaluate 3D object insertion for other vision tasks that requires accurate 3D ground-truth information.

III. Methods

After reviewing related work in synthetic image data generation, we chose to design a 3D object insertion-based approach for road scene generation. We first obtain the desired background image's scene appearance parameters: the road plane equation, vehicle trajectories, intrinsic and extrinsic camera parameters, and sun direction. Next, we use physics-based rendering to render 3D vehicle models with the obtained scene parameters. We also produce the corresponding, unoccluded depth maps for each vehicle.

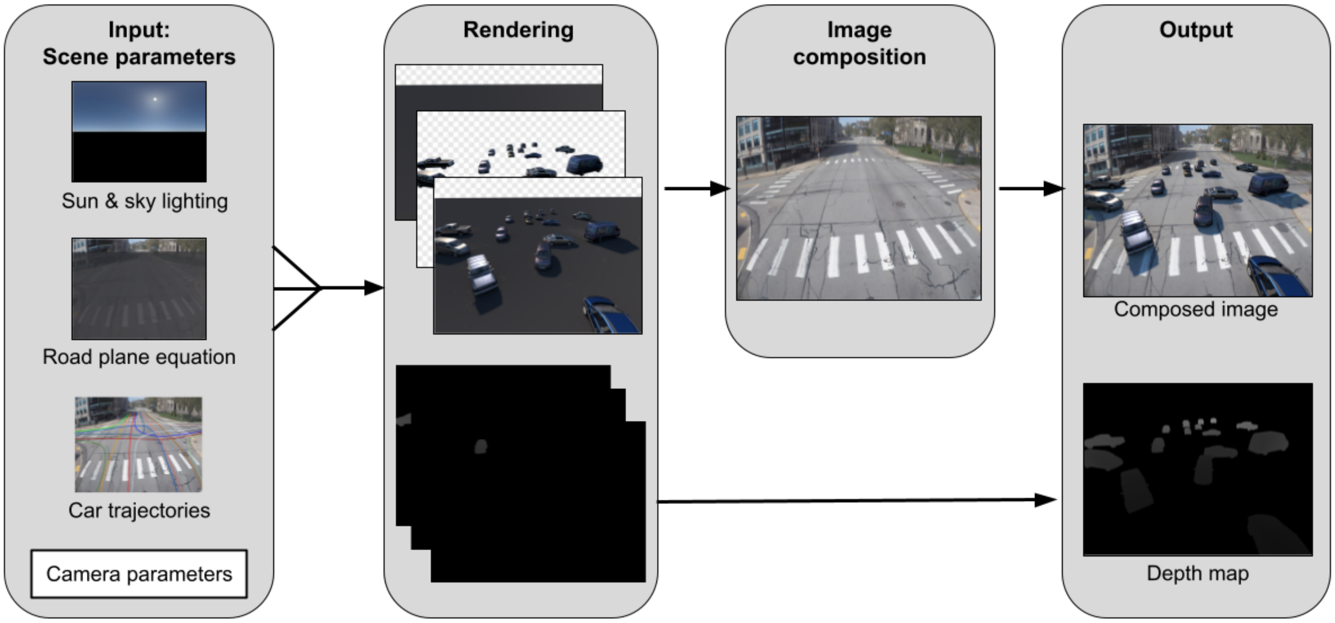


Fig. 1: Our proposed synthetic scene generation method

Finally, we composite the rendered cars onto the desired background using pixel-wise computations as described in [8]. Our synthetic scene generation method is illustrated in Fig. 1.

A. *Incorporating Scene Geometry*

To incorporate the desired background image's scene geometry into our vehicle renderings, we obtain the road plane equation, possible vehicle trajectories (Fig. 2), and intrinsic and extrinsic parameters of the camera that captured the desired background image. Note that our estimated scene geometry is in metric scale, thus allowing physically accurate renderings of objects.

Given the camera's GPS location, we leverage Google Street View (GSV) [9] to build the scene's geometry at that location. GSV is a street-level imagery database and a rich source of millions of panorama images with wide coverage all over the world. Every panorama image is geo-tagged with accurate GPS coordinates, capturing 360° horizontal and 180° vertical field-of-view with high resolution. We sample multiple panoramas around the desired camera's location inside a radius of 40 meters and use *structure-from-motion* (SfM) [10] to reconstruct the scene. Note that we also geo-registered the *up-to-scale* SfM reconstruction using the provided GPS coordinates of the GSV panoramas. Thus, our final 3D reconstruction of the scene is in *metric scale*.

To obtain the camera's intrinsic and extrinsic parameters, we follow the typical visual localization pipeline by localizing the desired background image (i.e., query image) w.r.t. the 3D reconstruction built with GSV images (i.e., database images). To establish robust 2D-3D correspondences, we follow hloc [11] by using learned feature matching method SuperGlue [12] with SuperPoint [13] features descriptors to match the query image with the database images. Given the 2D-3D correspondences, we perform a bundle adjustment step to retrieve the camera intrinsic and its 6DoF extrinsic parameters. Note that the large number of accurate matches between the query image and the rich GSV database images, produced by SuperPoint and SuperGlue, allows us to robustly recover both intrinsic and extrinsics parameters of the camera.

The road plane equation is estimated by fitting a plane to the set of 3D points whose 2D pixel locations are lying on the *road* obtained from off-the-shelf semantic segmentation method [14]. The possible vehicle trajectories are estimated from the real data by tracking multiple vehicles during a long period of time in 2D, which is then lifted to 3D using the road plane estimated above. We then perform spline-fitting followed by hierarchical clustering [15], where the average direction of each cluster is considered a possible vehicle trajectory. Additionally, when placing each 3D vehicle model into the scene along the vehicle trajectories, we employ collision checking between all models' 3D bounding boxes to ensure no models intersect each other in an unrealistic manner.

By incorporating the physically accurate scene geometry, as long as the inserted objects have correct metric scale, we are able to render geometrically accurate, scale-consistent road scenes (Fig. 3) and avoid inconsistencies such as unreasonable object sizes, incorrect distortions, or occlusions in our generated road scenes.

B. *Lighting Estimation*

We also estimate the environment map of the desired background image. After obtaining the time, date, and GPS coordinates for when and where the road scene photo was captured, our rendering software, Mitsuba [16], computes the sun's direction and generates environment map using sun

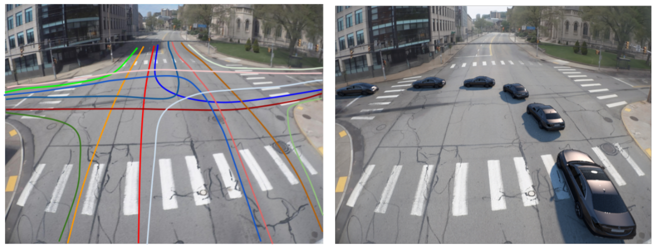


Fig. 2: All possible vehicle trajectories for this example road scene photo are visualized on the left. An illustration of one possible vehicle trajectory is generated on the right.

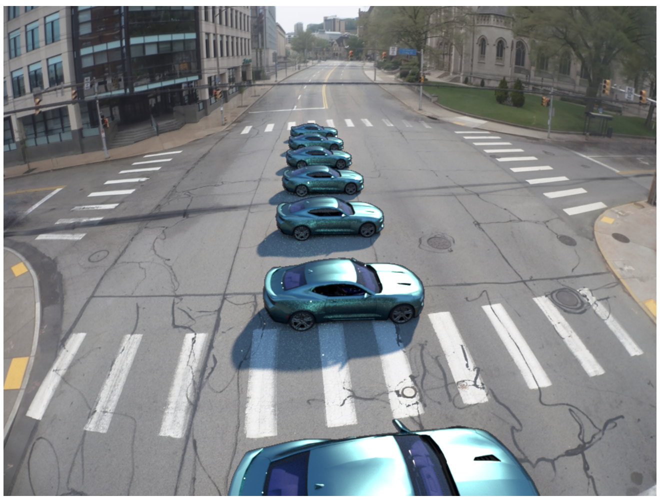


Fig. 3: Demonstration of our method's geometry, perspective, and size consistency as a result of using an estimated road plane equation and camera parameters. The same car model has been rendered and inserted at constantly increasing distances from the camera.

and sky illumination models. To avoid modeling the sun as a point light source and ensure that specular reflections are appropriately sized, we set Mitsuba's sun radius parameter to 5.

C. Physically-Based Rendering

We chose to render 3D objects using Mitsuba 0.5.0 [16] because it accurately models the physics of light scattering and can easily provide the corresponding, high resolution, ground truth rendering data such as depth maps, albedo maps, surface normals, and 3D coordinates in the world space. Other physically-based renderers also exist [8], [17], [18]. While these renderers are optimized and GPU-accelerated to be much faster than Mitsuba 0.5.0, their drawbacks include not being open-source, requiring RTX GPUs, and/or lacking crucial options that Mitsuba 0.5.0 provides. The specific options we require for our method's current implementation are the sun and sky illumination modeling plugin and the option to hide directly visible emitters.

Another vital part of physically-based rendering is incorporating the appropriate surface-scattering models for each type of material present in the scene. To achieve this, we

Material Names	Mitsuba Surface Scattering Models
"glass", "windshield"	Thin dielectric material
"plastic", "headlight", "indicator"	Smooth plastic material
"car body", "chrome", "metal", "silver"	Rough conductor material with Smooth dielectric coating
"tire", "rubber", "wheel"	Rough diffuse material
"interior"	Modified phong BRDF

Fig. 4: A summary of our material mapping rules. For each material name, we search the name for keywords and related substrings (left column), then match it to the appropriate Mitsuba surface scattering model (right column).

first curate a set of 11 high quality 3D vehicle models covering 5 categories (SUV, sedan, mini-van, van, pickup truck). Each 3D models must have a high polygon count and meaningful material names in its material file. Then, based on the material names, we can map each of the material definitions to the appropriate Mitsuba BRDF (bidirectional reflectance distribution function) surface scattering model. Our material mapping rules are summarized in Fig. 4.

D. Image Composition

We use an image composition method described in [8] to insert the 3D objects while blending their shadows into the background image. We render the following images for each road scene to obtain the necessary images for final image composition and depth maps:

- I_{all}: Road plane and 3D car models
- I_{obj}: 3D car models only
- I_{pl}: Road plane only
- Individual depth map for each 3D car model

All images are rendered with Mitsuba's options to hide directly visible emitters (in our case, the environment map) and enable the image's alpha channel. Then, the masks M_{all} and M_{obj} for I_{all} and M_{obj}, respectively, are easily obtained.

To remove potential pixel artifacts on object and plane edges, we erode the boundaries of regions of foreground pixels. Next, we calculate the edges, or contours, of the object mask M_{obj} and apply Gaussian blurring. Then, we alpha blend the blurred contours with the original contours.

Finally, to compute the new, composited image I_{new}, we calculate the pixel values in the object region by pixel-wise multiplication (indicated by $\odot$):

$$I_{new} \odot M_{obj} = I_{all} \odot M_{obj} \quad (1)$$

For the pixel values in the plane region:

$$I_{new} \odot (M_{all} - M_{obj}) = I \odot \frac{I_{all}}{I_{pl}} \odot (M_{all} - M_{obj}) \quad (2)$$

IV. RESULTS

We present image examples of results produced from our synthetic road scene generation method (Fig. 5, Fig. 6).

Note in each generated result, the rendered cars respect the scene lighting and geometry while exhibiting physically accurate reflections. Additionally, there are very few visible differences between the synthetic images and real world photos.

Fig. 5: An image example produced using our augmented reality-based data generation method. All cars in the image were rendered and inserted.

Fig. 6: An example of a real traffic camera photo, with one rendered vehicle inserted for comparison (the black SUV second from the bottom of the image)

For our rendering settings, we chose a volumetric path tracer in order to handle the relatively glossy car materials. We chose to render 1000×750 images at a sample count of 32 with a maximum path depth of 4 to balance the tradeoff between higher quality results and longer rendering times. In addition, we decided to render a random number between 10 and 20 vehicles, inclusive, to provide many vehicle examples in one image while maintaining reasonable rendering times.

Using these settings, we found that on one machine with 16 CPU cores, the rendering times for a single scene roughly vary between 200 and 400 seconds, with times largely being determined by the number of vehicles in the image. However, because one Mitsuba process will use at most 2GB of RAM for our method, multiple dataset generation and rendering processes can be launched simultaneously to produce a large synthetic image dataset.

V. CONCLUSION AND FUTURE WORK

The results demonstrate that our augmented reality-based method for synthetic road scene generation produces more photorealistic results compared to previous 3D object insertion works. Additionally, the results suggest that our approach can be readily used to generate road scene images and precise ground truth labels for computer vision tasks like object segmentation, construction zone detection, object tracking, 3D pose estimation, and more.

In future work, we plan to evaluate whether our synthetic data and precise depth maps (Fig. 7) improve training and performance of amodal segmentation models, which aim to predict the object's full segmentation mask despite visual occlusions [19]. We also plan to generate sets of traffic scenes that contain rare objects (Fig. 8) and object configurations (Fig. 9). We will evaluate whether adding these scenes helps improve object detection training and robustness.

Overall, these steps will allow us to discover how synthetic data generation can potentially augment existing vision datasets and train more robust computer vision models.

Fig. 7: We can compute the unoccluded segmentation masks from our output depth map and use them to train amodal segmentation models.

Fig. 8: Examples of rare objects: ambulance, firetruck, construction vehicle

Fig. 9: Example of a rare object configuration: Anomalous traffic pattern

ACKNOWLEDGMENT

This material is based upon work supported in parts by the National Science Foundation under Grants CCF-1730147 and CNS-2038612, and the Columbia University Egleston Scholars Program Stipend. Special thanks to the CMU Illumination and Imaging Lab and the Robotics Institute Summer Scholars program for their mentorship and support.

REFERENCES

[1] L. Niu, W. Cong, L. Liu, Y. Hong, B. Zhang, J. Liang, and L. Zhang, "Making images real again: A comprehensive survey on deep image composition," 2021. [Online]. Available: https://arxiv.org/abs/2106.14490

[2] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[3] Y. Wang, A. Liu, R. Tucker, J. Wu, B. L. Curless, S. M. Seitz, and N. Snavely, "Repopulating street scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[4] B. Yang, Y. Zhang, Y. Xu, Y. Li, H. Zhou, H. Bao, G. Zhang, and Z. Cui, "Learning object-compositional neural radiance field for editable scene rendering," in *International Conference on Computer Vision (ICCV)*, October 2021.

[5] J. Ost, F. Mannan, N. Thuerey, J. Knodt, and F. Heide, "Neural scene graphs for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 2856–2865.

[6] H. Abu Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented reality meets computer vision: Efficient data generation for urban driving scenes," *Int. J. Comput. Vision*, vol. 126, no. 9, p. 961–972, September 2018. [Online]. Available: https://doi.org/10.1007/s11263-018-1070-x

[7] Y. Chen, F. Rong, S. Duggal, S. Wang, X. Yan, S. Manivasagam, S. Xue, E. Yumer, and R. Urtasun, "Geosim: Realistic video simulation via geometry-aware composition for self-driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 7230–7240.

[8] Z. Li, M. Shafiei, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker, "Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2475–2484.

[9] Google, "Google Street View," https://www.google.com/streetview/.

[10] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *CVPR*, 2019.

[12] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[13] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[14] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," 2022.

[15] F. Li, N. D. Reddy, X. Chen, and S. G. Narasimhan, "Traffic4d: Single view reconstruction of repetitious activity using longitudinal self-supervision," in *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2021, pp. 1385–1392.

[16] W. Jakob, "Mitsuba renderer," 2010, http://www.mitsuba-renderer.org.

[17] "Marmoset Toolbag 4 - 3D Rendering, Texturing, & Baking Tools." [Online]. Available: https://marmoset.co/toolbag/

[18] "Omniverse Platform for Virtual Collaboration." [Online]. Available: https://www.nvidia.com/en-us/omniverse/

[19] N. D. Reddy, R. Tamburo, and S. G. Narasimhan, "Walt: Watch and learn 2d amodal representation from time-lapse imagery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 9356–9366.

Control Strategy Design for Bridge Painting using Image-based Visual-servoing on a Fully-Actuated UAV

Guanqi He[1], Junyi Geng[2], and Sebastian Scherer[2]

***Abstract*— Bridge painting is an essential work in urban maintenance. However, so far most of these works are mainly carried out by humans, which have the disadvantages of low efficiency, inconsistent quality, and issues of safety and health for human. A new vision-based hybrid motion and force controller is proposed for fully-actuated aerial vehicles to automate the bridge-painting work. Most of related interactive force control schemes for fully-actuated multi-rotors are designed for an indoor setting, where the position and velocity are measured with extra setup, i.e. motion capture system. However, in practical outdoor situations, the global position measurement can hardly be acquired. To tackle this challenge, we take advantage of image based visual-servo motion control strategy: recognize image features and directly compute control output via the pixel error rather than pose or velocity feedback. This approach does not rely on additional external sensors, such as RTK, GPS or ground station but only a minimal set of local sensors. Wall painting simulations is conducted to illustrate the performance of the proposed control scheme.**

***Index Terms*— Aerial Systems: Applications, Force Control, Motion Control, Visual Servoing**

I. INTRODUCTION

Bridge painting is an essential work in urban maintenance. Currently most the bridge painting works are completed manually, which suffers from low efficiency and safety risks. Also, the manual approach has the problem of increasing manpower costs and shortage of skilled workers at present, leading to the decline in the painting quality. Apart from painting by workers, some introduce robotic arm to complete bridge polishing and painting tasks. However, because robotic arms are fixed on the ground and lack of mobility, it is challenging for traditional manipulators to flexibly paint the bridge surface.

In this paper, we propose an our-door setting bridge-painting strategy using Fully-Actuated UAV. In the outdoor situation, the robot can only rely on a minimal set of local sensors, like IMU and camera. To tackle this challenge, we develop a hybrid force and visual-servoing control scheme. The rest of the paper is developed as follow: Section II introduces related works about line-based visual-servoing and force control with multi-rotors; Section III gives an overview of the painting strategy; In section IV, multi-rotor system dynamics is described; Section V illustrates the details of controller design, including contact force controller, visual-servo controller and attitude controller. And finally, Section VI presents the simulation experiments and results.

[1] ShanghaiTech University, Shanghai, China hegq@shanghaitech.edu.cn

[2] The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA. {junyigen, basti}@andrew.cmu.edu

Fig. 1. A fully-actuated hexarotor used in this work

II. RELATED WORKS

A. Line-based visual-servoing

Line-based visual servoing has been widely discussed. In [1], Espiau and Chaumette derived the interaction matrix of visual-servoing with line tracking feature. In [2], Araar and Aouf proposed a classical image-based visual servoing (IBVS) approach for the real time navigation of a quadrotor tracking power lines. [3] proposed an approach that solves the problem of automatic selection of the threshold during line tracking.

According to [1], a 3-D line can be represented as the intersection of two planes:

$$\begin{cases} A_1X + B_1Y + C_1Z + D_1 = 0 \\ A_2X + B_2Y + C_2Z + D_2 = 0 \end{cases}$$

Its 2-D projection on the image plane is parameterized with $(\rho, \theta) \in [0, \infty) \times [0, 2\pi)$, as known as the Hough parameter. In [1], the interaction matrix between the line feature parameter and the camera frame velocity has been derived:

$$\begin{bmatrix} \dot{\theta} \\ \dot{\rho} \end{bmatrix} = L \begin{bmatrix} \mathbf{V} \\ \mathbf{\Omega} \end{bmatrix} \tag{1}$$

with

$$L = \begin{bmatrix} \lambda_\theta \cos\theta & \lambda_\theta \sin\theta & -\lambda_\theta \rho & -\rho\cos\theta & -\rho\sin\theta & -1 \\ \lambda_p \cos\theta & \lambda_p \sin\theta & -\lambda_p \rho & (1+\rho^2)\sin\theta & -(1+\rho^2)\cos\theta & 0 \end{bmatrix} \tag{2}$$

where $\lambda_\theta = (A_i \sin\theta - B_i \cos\theta)/D_i$ and $\lambda_p = (A_i\rho\cos\theta + B_i\rho\sin\theta + C_i)/D_i$. It is worth to mention that the 3-D line expression (1) is expressed in the camera frame, which means the parameters A_i, B_i, C_i and D_i are varying with the movement of the drone. Basically, we need to extract the target tracking line on the image then reconstruct the line parameters to calculate the time-varying interaction matrix L.

B. Accurate force control with multi-rotor

Accurately control of the interaction force between the end effect and the bridge surface is essential to keep high bridge-painting quality. [4] proposed an impedance controller to realize the force interaction between UAV and environment. On this basis, the accurate feedback of the force is added to achieve accurate force control. In [5], [6], an energy-tank based wrench control algorithm is proposed, which effectively control the interaction force with dynamic environment, and the stability of the scheme has been rigorously proved. In this paper, we adopt an impedance integral force controller based on RGBD camera depth feedback and force sensor to control the interaction force when brushing the bridge.

III. Strategy overview

In this section, we will present our bridge painting strategy. Precise motion control of the UAV in the outdoor environment are required in the bridge-painting. Due to the poor GPS signal near the bridge, the global position measurement can hardly be acquired during the painting process. To tackle this challenge, we directly use image features and pixel error to feedback and control the motion of the robot, that is, image based visual-servoing control.

We assume that the boundary edges of the painting surface can be detected if they appear in camera view. In the first step, the drone moves close to the bridge to a pre-selected starting point by GPS Here we assume that the drone selects the bottom right corner as the starting point. Then the drone starts to track the bottom edge and paints the bridge side from right to left. As the robot arrives to the other side, it tracks the vertical edge to move up and then switches literal moving direction to paint back. When the robot is painting, the new edge is forming between the painted and unpainted area, which serves as visual tracking signal for future painting process. For example, in figure 3, the drone will track the yellow line (bottom edge) first and paint the blue area. Then the drone will move up and track the red line (new formed painting edge) to paint back.

Fig. 2. visual-servoing based wall painting

Fig. 3. Sketch of bridge painting strategy

IV. Multi-rotor dynamics

The dynamics of the fully-actuated UAV is governed by

$$\begin{aligned} m\dot{\mathbf{V}} &= -m\mathbf{\Omega} \times \mathbf{V} + \mathbf{F} \\ \mathbf{J}\dot{\mathbf{\Omega}} &= -\mathbf{\Omega} \times \mathbf{J}\mathbf{\Omega} + \mathbf{M} \end{aligned} \tag{3}$$

where $\mathbf{F} = -T + mg\mathbf{R}^\top e_3$, $\mathbf{\Omega} \in \mathbb{R}^3$, and $\mathbf{V} \in \mathbb{R}^3$ denotes the angular and linear velocity of the vehicle with respect to body frame $\{\mathcal{B}\}$, m is the mass of the vehicle, $e_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$ is the unit vector, $\mathbf{J}$ is the body frame inertia, $\mathbf{F} \in \mathbb{R}^3$ and $\mathbf{M} \in \mathbb{R}^3$ are the general external force and torque vector which expressed in the body frame, $g \in \mathbb{R}$ is the gravity acceleration and $\mathbf{R} \in \mathrm{SO}(3)$ is the rotation matrix describing the gravity direction. What's more, three sets of reference frames are defined for further controller design and analysis: a body-fixed frame $\{\mathcal{B}\}$, a camera frame $\{\mathcal{C}\}$ and an end-effector frame $\{\mathcal{E}\}$.

V. Controller Design

We divide the objective of the bridge painting task into three sub-tasks: 1) force control orthogonal to the contact surface. 2) attitude control ensuring end-effector opposite to the contact surface 3) motion control in the plane parallel to the wall.

For interaction force control, we utilize force sensor to acquire accurate contact force feedback. For the attitude control, the normal vector of the contact surface is obtained by the depth camera to feedback the attitude of the robot. And visual-servoing line-tracking is used for motion control.

For simplicity, before we illustrate the detailed design of controller, let's set up a reference frame $\{\mathcal{W}\}$ on the working surface (the wall to be painted) first. Let $\mathbf{x}_\mathcal{W}$, $\mathbf{y}_\mathcal{W}$, $\mathbf{z}_\mathcal{W}$ be the X axis, Y axis and Z axis unit vector of frame $\{\mathcal{W}\}$, respectively, where $\mathbf{x}_\mathcal{W}$ is aligned with surface norm pointing into the contact surface, $\mathbf{y}_\mathcal{W}$ is aligned with line direction, and $\mathbf{z}_\mathcal{W} = \mathbf{x}_\mathcal{W} \times \mathbf{y}_\mathcal{W}$. The origin $O_\mathcal{W}$ is placed on the contact point. The transition matrix ${}^{\mathcal{W}}T_{\mathcal{C}}$ can be obtained through the surface normal vector ${}^{\mathcal{C}}\mathbf{n} \in \mathbb{R}^3$ with respect to camera frame and the line direction vector $l \in \mathbb{R}^2$ on image plane. By the definition of wall frame $\{\mathcal{W}\}$,

$${}^{\mathcal{C}}\mathbf{x}_\mathcal{W} = {}^{\mathcal{C}}\mathbf{n} \tag{4}$$

As line direction vector l is the projection of ${}^{\mathcal{C}}\mathbf{y}_{\mathcal{W}}$ on the image plane, we have

$$\begin{aligned}\frac{l_y}{l_x} &= \frac{{}^{\mathcal{C}}\mathbf{y}_{\mathcal{W}y}}{{}^{\mathcal{C}}\mathbf{y}_{\mathcal{W}x}} \\ {}^{\mathcal{C}}\mathbf{x}_{\mathcal{W}} \cdot {}^{\mathcal{C}}\mathbf{y}_{\mathcal{W}} &= 0 \\ \|{}^{\mathcal{C}}\mathbf{y}_{\mathcal{W}}\| &= 1\end{aligned} \tag{5}$$

${}^{\mathcal{C}}\mathbf{y}_{\mathcal{W}}$ can be obtained by solving the equation above, and ${}^{\mathcal{C}}\mathbf{z}_{\mathcal{W}} = {}^{\mathcal{C}}\mathbf{x}_{\mathcal{W}} \times {}^{\mathcal{C}}\mathbf{y}_{\mathcal{W}}$. Then the obtained results is applied to construct the rotation matrix ${}^{\mathcal{C}}\mathbf{R}_{\mathcal{W}} = \begin{bmatrix}\mathbf{x}_{\mathcal{W}} & \mathbf{y}_{\mathcal{W}} & \mathbf{z}_{\mathcal{W}}\end{bmatrix}$. Since the origin of wall frame and end-effector frame is overlapped, ${}^{\mathcal{C}}O_{\mathcal{W}} = {}^{\mathcal{C}}O_{\mathcal{E}}$. Next, the design of the force, attitude and visual servo controller will be discussed.

A. Force controller

One of the requirements of painting task is to keep the end-effecot touch with the contact surface and maintain a certain pushing force while the drone is moving. Here an explicit force controller is introduced to achieve stable force control. Define contact force error $\tilde{F}_w = F_{w,s} - F_{w,d}$, where $F_{w,s}$ denotes the contact force measured by the force/torque sensor and $F_{w,d}$ denotes the desired contact force. The explicit form of generated contact force F is as follow

$$\dot{F}_w = K_w \tilde{F}_w, \qquad F_w(0) = F_0 \in \mathbb{R} \tag{6}$$

where K_w is the tuning gain. Due to the control input $\mathbf{F}$ is defined and generated in body frame $\{\mathcal{B}\}$, we convert F_w from wall frame $\{\mathcal{W}\}$ to the body frame $\{\mathcal{B}\}$.

$$\begin{bmatrix}\mathbf{F}_w \\ \mathbf{M}_w\end{bmatrix} = {}^{\mathcal{B}}\mathbf{V}_{\mathcal{W}} F_w e_1 \tag{7}$$

where $e_1 = \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\end{bmatrix}^\top$ is the unit vector, ${}^{\mathcal{B}}\mathbf{V}_{\mathcal{W}}$ is the matrix that transforms velocities from the wall frame to the body frame.

$${}^{\mathcal{B}}\mathbf{V}_{\mathcal{W}} = \begin{bmatrix}{}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}} & 0_{3\times 3} \\ [{}^{\mathcal{B}}\mathbf{t}_{\mathcal{W}}]_\times {}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}} & {}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}\end{bmatrix} \tag{8}$$

where ${}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}}$ is the rotation matrix between the wall frame and the robot body, ${}^{\mathcal{B}}\mathbf{t}_{\mathcal{W}}$ is the corresponding constant translation vector and $[\mathbf{t}_\times]$ is the skew-symmetric matrix related to $\mathbf{t}$.

B. Contact attitude controller

Another important factor which effects the painting quality is the attitude of the end-effector during the painting process. The ideal condition is keeping the end-effector orthogonal to the contact surface . To achieve this goal, we use the normal vector $\mathbf{n}$ of contact surface obtained via RGB-D camera as the attitude feedback.

Here a simple state-feedback attitude controller is proposed. Define yaw angle error $e_\psi = \psi - \psi_d$, pitch angle error $e_\theta = \theta - \theta_d$ and $e_a = \begin{bmatrix}0 & 0 & 0 & 0 & e_\theta & e_\psi\end{bmatrix}^\top$, where relative angle measurements ψ and θ are calculated using surface normal vector.

$$\begin{aligned}{}^{\mathcal{E}}\mathbf{n} &= {}^{\mathcal{E}}R_{\mathcal{C}} {}^{\mathcal{C}}\mathbf{n} \\ \psi &= \arctan\left(\frac{{}^{\mathcal{E}}\mathbf{n}_y}{{}^{\mathcal{E}}\mathbf{n}_x}\right) \\ \theta &= \arctan\left(\frac{{}^{\mathcal{E}}\mathbf{n}_z}{\sqrt{{}^{\mathcal{E}}\mathbf{n}_x^2 + {}^{\mathcal{E}}\mathbf{n}_y^2}}\right)\end{aligned} \tag{9}$$

A PI controller is implemented to control the robot attitude orthogonal to the surface.

$$\begin{bmatrix}\mathbf{F}_a \\ \mathbf{M}_a\end{bmatrix} = {}^{\mathcal{B}}\mathbf{V}_{\mathcal{C}}\left(K_p e_a(t) + K_i \int_0^t e_a(\tau)d\tau\right) - K_d \begin{bmatrix}\mathbf{0}_3 \\ \mathbf{\Omega}\end{bmatrix} \tag{10}$$

where $\mathbf{0}_3 = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^\top \in \mathbb{R}^3$.

C. visual-servoing tracking controller

As discussed in previous sections, the drone needs to smoothly track the edge of the bridge or the line painted previous. Here we introduce a line-based IBVS controller to ensure the motion of end-effector is aligned with the tracking line.

Let $q = (\theta, \rho)^\top$ denotes the line parameter on the image plane, $q_s = (\theta_s, \rho_s)^\top$ denotes the tracking target and $e_q = q - q_s$ denotes the visual servoing error. The error dynamics reads as

$$\dot{e}_q = \dot{q} - \dot{q}_s \tag{11}$$

$$= L_{sc}\begin{bmatrix}\mathbf{V} \\ \mathbf{\Omega}\end{bmatrix} - \dot{q}_s \tag{12}$$

$$\ddot{e}_q = L_{sc}\begin{bmatrix}-\mathbf{\Omega} \times \mathbf{V} + m^{-1}\mathbf{F} \\ -\mathbf{J}^{-1}\mathbf{\Omega} \times \mathbf{J}\mathbf{\Omega} + \mathbf{J}^{-1}\mathbf{M}\end{bmatrix} - \dot{q}_s \tag{13}$$

The visual servoing controller is then given as below.

$$\begin{aligned}\begin{bmatrix}\mathbf{F}_{v0} \\ \mathbf{M}_{v0}\end{bmatrix} &= \begin{bmatrix}\mathbf{0}_3 \\ \mathbf{\Omega} \times \mathbf{J}\mathbf{\Omega}\end{bmatrix} + \\ L_{sc}^\dagger &\left[K_{v,p} e_q + K_{v,i}\int_0^t e_q(\tau)d\tau + K_{v,d}\dot{e}_q(t) + \dot{q}_s\right]\end{aligned} \tag{14}$$

where $L_{sc}^\dagger$ is the pseudo-inverse of matrix L_{sc} Consider the fact that the controller only focuses on roll angle ϕ and z-direction distance ${}^{\mathcal{W}}z$ in wall frame, define the selection matrix

$${}^{\mathcal{W}}\mathbf{S}_v = \mathbf{O}_{2\times 2} \oplus \mathbf{I}_{1\times 1} \oplus \mathbf{O}_{2\times 2} \oplus \mathbf{I}_{1\times 1} \in \mathbb{R}^{6\times 6} \tag{15}$$

Convert the selection matrix into body frame we have

$${}^{\mathcal{B}}\mathbf{S}_v = {}^{\mathcal{B}}\mathbf{R}_{\mathcal{W}} {}^{\mathcal{W}}\mathbf{S}_v {}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}} \tag{16}$$

Finally, the expression of visual servo controller is written as bellow

$$\begin{bmatrix}\mathbf{F}_v \\ \mathbf{M}_v\end{bmatrix} = {}^{\mathcal{B}}\mathbf{S}_v \begin{bmatrix}\mathbf{F}_{v0} \\ \mathbf{M}_{v0}\end{bmatrix} \tag{17}$$

D. General Controller Form

In the end, sum all three controllers proposed above, the general form of the controller is given below.

$$\begin{bmatrix}\mathbf{F}\\ \mathbf{M}\end{bmatrix} = \begin{bmatrix}\mathbf{F}_w\\ \mathbf{M}_w\end{bmatrix} + \begin{bmatrix}\mathbf{F}_a\\ \mathbf{M}_a\end{bmatrix} + \begin{bmatrix}\mathbf{F}_v\\ \mathbf{M}_v\end{bmatrix} \tag{18}$$

In our controller frame work, desired contact force $F_{w,d}$, desired tracking line parameter (ρ_d, θ_d) and desired attitude ψ_d, θ_d are serve as the input, and wrench $\{\mathbf{F}, \mathbf{M}\}$ are the output. The low-level firmware of the PX4-fully-actuated [7] allocates the force and torque to the propeller speed. The scheme of the proposed controller is shown in the diagram below Fig 4.

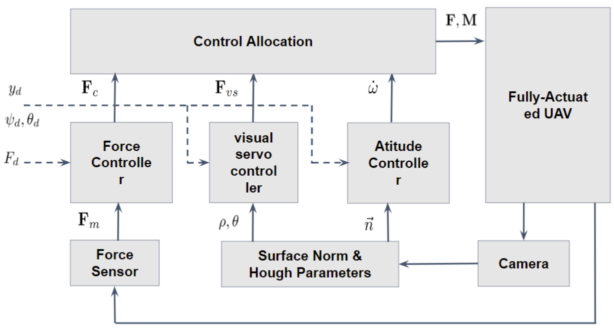


Fig. 4. Architecture of the wall-painting controller

VI. Experiments and Results

To assess the performance of the proposed control scheme, we model the fully-actuated UAV in both Gazebo simulator and Matlab Simulink based on the design described in [7]. The dynamic control allocation is developed on top of the PX4 source code, which can run directly on real aircraft.

In the simulation, the drone is initialized at the origin. The drone moves close to the bridge using GPS guidance, then starts to paint 8 meters bridge side wall. Simulation results show that the proposed approach succeeded in tracking the bottom line, maintaining a stable attitude and desired contact force. Figure 6 presents the 3D motion trajectory of the drone with visual-servo controller. For visual-servo controller (Figure 7), the average tracking error is less than 0.04 (about 1.6 cm) , providing a efficient height control. For force controller, Figure 9 shows that average force tracking error is within 0.11N, ensuring stable wall painting.

VII. Conclusion

In this paper, a novel image-based visual-servoing bridge-painting scheme for fully-actuated UAV is proposed. The scheme consists of three controllers: visual-servoing height controller, surface norm based attitude controller and impedance integral contact force controller. Simulation results present the validity of the proposed approach.

In the future, the algorithm will be migrated to the hardware platform and tested in both indoor and outdoor

Fig. 5. Wall-painting in Gazebo

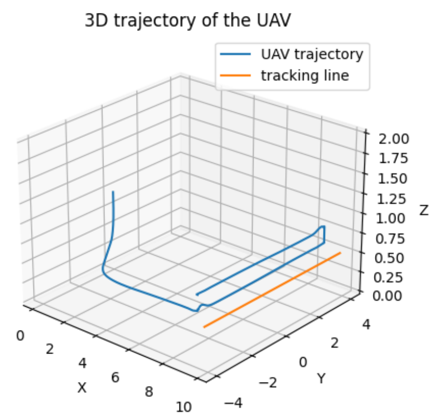


Fig. 6. 3D trajectory of the painting task

environment. What's more, the velocity control on the tracking direction ${}^{W}Y$ only takes IMU as the feedback, which suffers from integral drift. Next step we will add an optical flow controller on this direction, using sparse visual feature to achieve velocity close-loop control.

References

[1] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *ieee Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1992.

[2] O. Araar and N. Aouf, "Visual servoing of a quadrotor uav for autonomous power lines inspection," in *22nd Mediterranean Conference on Control and Automation*. IEEE, 2014, pp. 1418–1424.

[3] G. Zhou, J. Yuan, I.-L. Yen, and F. Bastani, "Robust real-time uav based power line detection and tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 744–748.

[4] V. Lippiello and F. Ruggiero, "Cartesian impedance control of a uav with a robotic arm," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 704–709, 2012.

[5] M. Brunner, L. Giacomini, R. Siegwart, and M. Tognon, "Energy tank-based policies for robust aerial physical interaction with moving objects," *arXiv preprint arXiv:2202.06755*, 2022.

[6] R. Rashad, D. Bicego, J. Zult, S. Sanchez-Escalonilla, R. Jiao, A. Franchi, and S. Stramigioli, "Energy aware impedance control of a flying end-effector in the port-hamiltonian framework," *IEEE Transactions on Robotics*, pp. 1–20, 2022.

[7] A. Keipour, M. Mousaei, A. Ashley, and S. Scherer, "Integration of fully-actuated multirotors into real-world applications," pp. 1–8, in press.

2014, pp. 2113–2118.

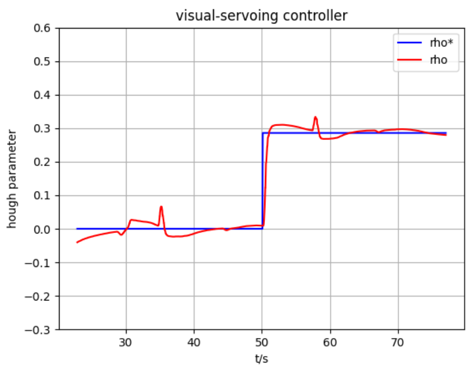


Fig. 7. Measured and desired tracking line parameter(ρ)

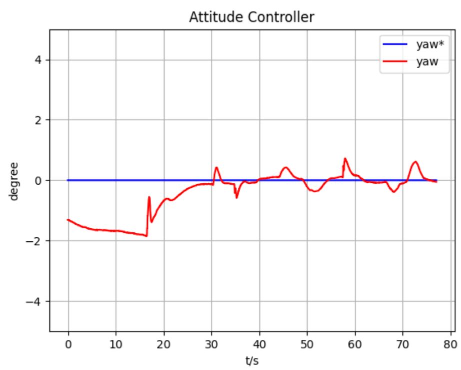


Fig. 8. Measured and desired yaw(ψ) angle

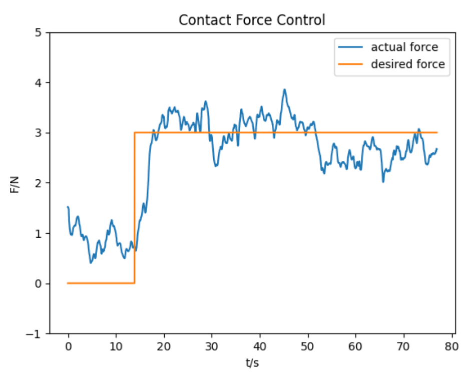


Fig. 9. Measured and desired contact force F_w

[8] A. Keipour, M. Mousaei, A. T. Ashley, and S. Scherer, "Integration of fully-actuated multirotors into real-world applications," *arXiv preprint arXiv:2011.06666*, 2020.

[9] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The confluence of vision and control*. Springer, 1998, pp. 66–78.

[10] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[11] P. Couvignou and P. Khosla, "Singularities in the recovery of rigid body motions from optical flow data," Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-92-20, October 1992.

[12] J. Thomas, G. Loianno, K. Sreenath, and V. Kumar, "Toward image based visual servoing for aerial grasping and perching," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE,

Introspective, Explainable Action Advising

Gus Brocchini[1], Yue Guo[2], Joseph Campbell[2], Simon Stepputtis[2], and Katia Sycara[2]

***Abstract*— Action advising on a budget is a common framework for transfer reinforcement learning. Recent work [1] has extended action advising to provide an explanation along with the action so that advice can be reused, improving performance on the same advice budget. However, often in transfer learning, we want to use a teacher that has been trained on a different but related task to the student's. In this case, advice reuse could stop the student from visiting preferable states to the ones the teacher tends towards, decreasing performance. We propose a framework for teacher introspection that allows the teacher to avoid or retract bad advice. By treating the student's experience as off-policy RL data for the teacher, we calculate the temporal difference (TD) error according to the teacher's value function. If the states associated with a certain explanation have a large, consistent TD error, the teacher can avoid giving that explanation, or retract it if already given.**

***Index Terms*— Deep Learning Methods, Reinforcement Learning, Transfer Learning,**

I. INTRODUCTION

A central problem in machine learning is sample inefficiency; it often takes artificial agents orders of magnitude more experience than human agents to achieve comparable performance on a task. If we already have an agent that performs well at a similar task, we can leverage that expert agent's knowledge to teach a new agent, thereby accelerating learning. [2] In reinforcement learning, a standard method for transferring knowledge between agents is action advising on a budget, where the teacher recommends actions to the student, constrained by a communicatioin budget. [3]

Part of the reason that action advising is effective in reinforcement learning is because the student's training data is dependent on its performance. If the student behaves more like an expert early in training, it is more likely to develop experience—and therefore good decision making—in the areas it will encounter when it becomes an expert at its task. One can imagine the case of a self-driving car; without action advising, the student is likely to spend the early stages of training learning effective strategies for how to get back on the road after having driven off. Of course, this knowledge will largely be wasted once it learns to simply stay on the road. Ideally, action advising lets us skip learning these intermediate skills that are primarily needed for non-expert agents.

A limitation of action advising, when compared to human teaching and learning, is the inability to reuse advice except when the *exact* same state occurs. [1], [4] Several methods have been proposed to address this. In this paper, we extend the method proposed in [1], which provides an explanation with the advice. If this explanation fits for other states, the student can reuse the advice.

The most common motivating factor for using student-teacher interactions—with some exceptions, notably if either agent is a human or if a different policy model is preferable for some reason—is that the teacher and the student are trained on distinct but related tasks. In this transfer learning scenario, there is some theoretical danger to [1]'s explainable action advising. Because the teacher is not an expert agent, in certain states, the teacher will systematically bias the student *away* from expert states instead of *towards* them.

So, we must somehow avoid giving bad advice. The naïve approach is to use the student's value function to evaluate the teacher's advice. This, however, depends on the student's value function being approximately accurate, and once the student achieves that level of performance it is better to ignore the teacher's advice altogether. In other words, the teacher is only useful when the student is bad. Therefore, we cannot rely on the student's model at all.

Instead, we can use the student's experience as off-policy RL data to evaluate the teacher's policy. At each timestep, we use the teacher's Q function to calculate its TD error, giving a measure of how well the teacher predicts the environment's behavior. The TD error has the advantage of directly measuring the teacher's knowledge of the student's task—even if the student is experiencing states very different from the teacher's, the teacher will give advice if its error is low. These errors can be aggregated by the leaves of the decision tree, thereby giving a score for each explanation. The teacher can avoid giving explanations that have large errors, and can even retract past explanations if it realizes they may be wrong.

II. BACKGROUND

A. Reinforcement Learning

Reinforcement learning describes a class of methods for training an artificial agent on decision making problems. The task and environment are modeled as a Markov Decision Process (MDP), which is a tuple (S, A, R, T, γ), where S is the set of states, A is the set of actions, $R : S \times A \times S \to \mathbb{R}$ is the reward function, $T : S \times A \times S \to \mathbb{R}$ defines the state transitions and $\gamma \leq 1$ is the discount factor. At each timestep t, the agent recieves a state s_t and performs an action a_t, transitioning to state s_{t+1} and recieving reward r_t. Actions are determined by a policy $\pi : S \to A$ and the agent optimizes the total discounted reward $\sum_t \gamma^t r_t$.

[1]Gus Brocchini is with the Department of Computer Science, Yale University, New Haven, CT, USA gus.brocchini@yale.edu

[2]Yue Guo, Joseph Campbell, Simon Stepputtis, and Katia Sycara are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA {yueguo,jcampbell,stepputtis,katia}@cmu.edu

B. Action Advising

In reinforcement learning, action advising on a budget has been proposed as a model-agnostic method for transfer learning. [3] In action advising, the student announces its observed state and the action it intends to take at each time step. Then, the teacher decides (using a heuristic function) whether or not to recommend a different action to take instead. The teacher is constrained by an "advice budget"—the number of times it is allowed to issue advice. Action advising has the significant benefit of not making any assumption about the architectures of either the teacher or student; they need only share an action set. This makes teaching across architectures—or even between humans and artificial agents—easy.

There are several common heuristic functions used to determine whether or not to issue advice. [3] The simplest is early advising, where the teacher issues advice at every timestep until the advice budget runs out. Another heuristic is importance advising, where the teacher uses its Q function to calculate the importance of a state s as $I(s,a) = \max_a Q(s,a) - \min_a Q(s,a)$ and gives advice if this importance crosses a certain threshold. In mistake correcting, the teacher gives advice only if a state is important enough and the student's announced action differs from the teacher's recommended action.

III. Related Work

A. Advising under uncertainty

Some work has been done on action advising under uncertainty, especially in the multi-agent case. In [5], agents learn and teach each other simultaneously, using the number of times a state has been visited as a proxy for confidence in prediction. This works best with a small, discrete state space, although it can be adapted for continuous environments by segmenting the state space into regions. The main drawback of this work is that it relies on the students and teachers sharing the same environment. In our scenario, the number of times the teacher has visited a state has no bearing on whether that state behaves simlilarly in the student's environment.

In [6], agents, in addition to learning to perform their task, learn to ask for and give advice. This is a theoretically robust method for teaching and learning, but introduces complexity with a second learning step, and is again tailored for the case where all agents share a single environment.

In [7], the student requests advice when it is highly uncertain about what action to take. This does not translate well to our task-transfer scenario, where the teacher, having converged on its original task, is likely to be very confident in its predictions regardless of whether they are correct for the student's task.

B. Advice Reuse

In [4], the student's exploration method is modified to occaisionally reuse previously recieved advice using imitation learning. This makes the connection between action advice and better exploration explicit. The authors use early advising which allows the student to collect a full dataset of teacher advice before using behavioral cloning to imitate it.

[8] proposes several methods for the student to decide whether to explore, reuse advice, or follow its own actions. These methods include using the student's value function to evaluate past advice; following the advice a set number of times and then forgetting it; and a decaying reuse probability.

In [1], an explanation, in the form of a path down a decision tree, is provided with the action advice so that the student can generalize advice to similar states. This leads to better performance on the same advice budget as the student is able to use the teacher's advice in more situations, and is the basis for this work. The student reuses advice with a decaying probability, as in [8].

IV. Methodology

Algorithm 1 Introspective Explainable Action Advising

Inputs: Teacher value function Q, teacher policy π^*, heuristic function h. **Parameters:** Advice budget b, memory decay rate $\lambda < 1$, error threshold e.

Distilled decision tree $\hat{\pi}^* = VIPER(\pi^*)$
Reconstructed tree $\pi' = \emptyset$
Initialize student policy π
Initialize TD error aggregate E to 0 for all paths on $\hat{\pi}^*$
for iter $i = 1, 2, \dots$ **do**
  for timestep $t = 1, \dots, T$ **do**
    $\delta = Q(s_{t-1}, a_{t-1}) - r_{t-1} - \gamma Q(s_t, a_t)$
    update E with $\delta, \texttt{path}(\hat{\pi}^*, s_{t-1})$
    if $s_t \in \pi'$ **then**
      with probability λ^i: take $\pi'(s_t)$
      otherwise: take $\pi(s_t)$
    else if $b > 0$ and $h(s_t)$ and $|E(s_t)| < e$ **then**
      give advice $a_t = \hat{\pi^*}(s_t)$
      reconstruct tree $\pi' = \pi' \cup \texttt{path}(\hat{\pi}^*, s_t)$
      $b = b - 1$
    else
      take student action $\pi(s_t)$
    end if
  end for
  for path p in $\hat{\pi}^*$ **do**
    if $b > 0$ and $p \in \pi'$ and $|E(p)| > e$ **then**:
      $\pi' = \pi' \setminus p$
      $b = b - 1$
    end if
  end for
  update π according to policy optimization
end for

The algorithm for IE-AA is derivative of the original E-AA algorithm from [1]. It takes the teacher's policy and value function, along with parameters for advice budget and the rate of memory decay. The error threshold is the threshold above which advice will be retracted. The first step of the algorithm is to distill the teacher's policy into a decision tree using VIPER, as described in [9]. Then, the algorithm runs largely the same as E-AA, reusing advice with a certain probability dependent on the memory decay rate, and giving advice when the heuristic function says to.

The differences are in that each timestep, the running average TD error E is updated, and if E is greater than the threshold e, advice is not given. Then, at the end of each episode, if any previously given advice now has error over the threshold, the teacher retracts the advice. (This also decrements the advice budget, as the budget is meant to constrain communication between the teacher and student.)

V. EXPERIMENTAL SETUP

Due to difficulties in training and a desire to get at least preliminary results for this paper, we used a simplified 8x8 gridworld environment. The students were trained on an environment that had a wall with a one-unit gap between the agent and the goal, and the teacher was trained on the same environment without the wall. An exploration bonus in the form of a penalty for shortest-path distance to the goal was included to speed up training.

Also for simplicity, we represented the state symbolically. The location of the opening in the wall and the goal were represented relative to the position and direction of the agent, and two additional variables were included to indicate whether the agent was past the opening in the wall and whether the tile directly in front of the agent is a wall. The decision tree was trained on these same features.

VI. RESULTS

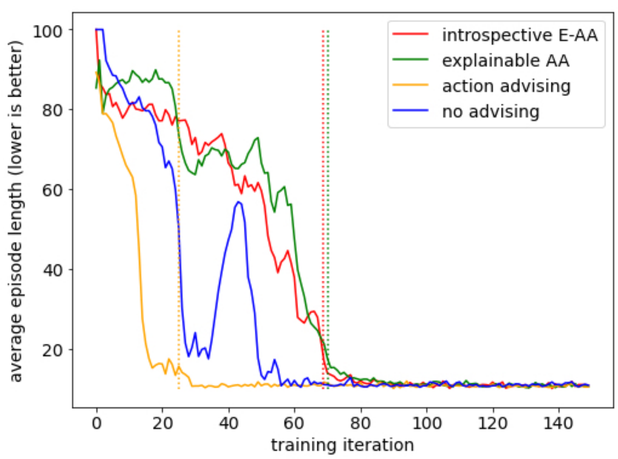


Fig. 1. Preliminary results showing average episode length for four different advising schemes. The heuristic function used for all schemes was alternating advising with an interval of 5. Dotted lines indicate when the advising budget ran out.

At this time, we only have extremely preliminary results. These results, using alternating advising with an interval of 5, demonstrate that explanations can in fact lead to decreased performance compared to standard action advising, or even no advising. The results from introspection are inconclusive, as the introspection parameters have not been tuned.

ACKNOWLEDGMENT

This work was supported by the Robotics Institute Summer Scholars program. A special thanks goes to the directors of the program, Rachel Burcin and Dr. John M. Dolan, for their direction.

REFERENCES

[1] Y. Guo, J. Campbell, S. Stepputtis, R. Li, D. Hughes, F. Fang, and K. Sycara, "A teacher-student policy transfer framework for giving explainable action advice in multi-agent reinforcement learning," 2022.

[2] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[3] L. Torrey and M. Taylor, "Teaching on a budget: Agents advising agents in reinforcement learning," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 2013, pp. 1053–1060.

[4] E. Ilhan, J. Gow, and D. Perez-Liebana, "Action advising with advice imitation in deep reinforcement learning," *arXiv preprint arXiv:2104.08441*, 2021.

[5] F. L. Da Silva, R. Glatt, and A. H. R. Costa, "Simultaneously learning and advising in multiagent reinforcement learning," in *Proceedings of the 16th conference on autonomous agents and multiagent systems*, 2017, pp. 1100–1108.

[6] S. Omidshafiei, D.-K. Kim, M. Liu, G. Tesauro, M. Riemer, C. Amato, M. Campbell, and J. P. How, "Learning to teach in cooperative multi-agent reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6128–6136.

[7] F. L. Da Silva, P. Hernandez-Leal, B. Kartal, and M. E. Taylor, "Uncertainty-aware action advising for deep reinforcement learning agents," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 04, 2020, pp. 5792–5799.

[8] C. Zhu, Y. Cai, H.-f. Leung, and S. Hu, "Learning by reusing previous advice in teacher-student paradigm," in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, 2020, pp. 1674–1682.

[9] O. Bastani, Y. Pu, and A. Solar-Lezama, "Verifiable reinforcement learning via policy extraction," *Advances in neural information processing systems*, vol. 31, 2018.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

Leveraging Structure from Motion to Localize Inaccessible Bus Stops

Indu Panigrahi[1], Tom Bu[2], and Christoph Mertz[2]

***Abstract*— The detection of hazardous conditions near public transit stations is necessary for ensuring the safety and accessibility of public transit. Smart city infrastructures aim to facilitate this task among many others through the use of computer vision. However, most state-of-the-art computer vision models require thousands of images in order to perform accurate detection, and there exist few images of hazardous conditions as they are generally rare.**

In this paper, we examine the detection of snow-covered sidewalks along bus routes. Previous work has focused on detecting other vehicles in heavy snowfall or simply detecting the presence of snow. However, our application has an added complication of determining if the snow covers areas of importance and can cause falls or other accidents (e.g. snow covering a sidewalk) or simply covers some background area (e.g. snow on a neighboring field). This problem involves localizing the positions of the areas of importance when they are not necessarily visible.

We introduce a method that utilizes Structure from Motion (SfM) rather than additional annotated data to address this issue. Specifically, our method learns the locations of sidewalks in a given scene by applying a segmentation model and SfM to images from bus cameras during clear weather. Then, we use the learned locations to detect if and where the sidewalks become obscured with snow. After evaluating across various threshold parameters, we identify an optimal range at which our method consistently classifies different categories of sidewalk images correctly. Although we demonstrate an application for snow coverage along bus routes, this method can extend to other hazardous conditions as well. Code for this project is available at `https://github.com/ind1010/SfM_for_BusEdge`.

***Index Terms*— Computer Vision for Transportation, Intelligent Transportation Systems, Localization, Segmentation and Categorization**

I. INTRODUCTION

Smart city infrastructures aim to use fields like computer vision to facilitate city management, part of which involves overseeing transportation systems. As transportation systems become more intelligent, an increasing amount of public transit vehicles are equipped with cameras that capture thousands of images of the city per day along with geographic positioning information. City infrastructures can use this immense amount of raw data to monitor the conditions of public transit stations and the surrounding areas.

Our application focuses on detecting snow-covered sidewalks along bus routes; snow-covered sidewalks are one type of hazardous condition that can limit the safety and accessibility of public buses as pedestrians can lose access to bus stops and/or slip (Fig. 1). We use images that are captured on-board a public bus as data. However, instead of annotating this data, we leverage the fact that the bus travels around a set route and apply Structure from Motion and a segmentation model to learn the locations of the sidewalks in clear weather. Then, in future rounds, when the bus encounters snowfall, we compare the detected snow coverage to the learned locations of the sidewalks. If the coverage exceeds a set threshold, we generate an alert, and the bus company can contact the city to clear the sidewalk.

Fig. 1: Snow-covered sidewalk leading to a bus stop.

When evaluating on a few categories of sidewalk images, we identify a set of thresholds at which our method performs well across all categories for this bus route. Though we demonstrate an application for detecting snow-covered sidewalks, our method can generalize to detecting other conditions such as snow on roads or bike lanes.

Our contributions are as follows:

- We present a method that combines Structure from Motion with a segmentation model to learn the expected locations of sidewalks and detect whether or not the learned sidewalk locations become covered by snow.
- Although we demonstrate by detecting snow-covered sidewalks, our method can easily generalize to other problems.
- We collect a small dataset of images depicting sidewalks in clear and snowy weather that we use for evaluation. Additionally, we compile other categories of images that may be relevant for other works.

II. RELATED WORK

A. *Existing Municipal Infrastructures*

Many American cities use the telephone number 311 that allows anyone to report issues for the city to fix, such as snow-covered sidewalks. However, this process can be inefficient as it is decentralized and relies on the motivation of people.

[1]Indu Panigrahi is with Robotics Institute Summer Scholars Program at Carnegie Mellon University, Pittsburgh, PA 15213, USA and also with the Department of Computer Science at Princeton University, NJ 08544, USA `indup@princeton.edu`

[2]Tom Bu and Christoph Mertz are with the Robotics Institute at Carnegie Mellon University, Pittsburgh, PA 15213, USA `tomb, cmertz@andrew.cmu.edu`

B. *BusEdge*

Since buses regularly travel around cities, and many are equipped with cameras, we can facilitate the detection of municipal problems by regularly analyzing bus camera images. We use a platform called BusEdge [1] that captures and packages images with GPS information from the client (bus) and sends the data to the server (cloudlet) to be analyzed (Fig. 2). Intensive on-board analysis can be limited because the bus is equipped with a CPU.

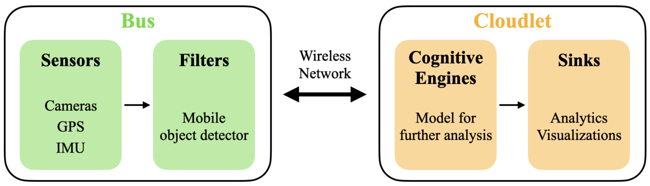


Fig. 2: Overview of BusEdge Platform. Figure adapted from Fig. 3.1 in [1].

C. *Panoptic Segmentation*

Panoptic segmentation combines semantic and instance segmentation by both categorizing pixels that represent uncountable areas (e.g. snow) and grouping pixels into instances if they belong to countable objects (e.g. cars) [2]. Although our application involves semantic segmentation categories, we employ a panoptic segmentation model so that our method can be extended more easily for applications where instances are needed.

We apply an off-the-shelf segmentation model called Mask2Former [3]. This model incorporates a Transformer decoder. Transformers have recently become a popular option for computer vision models in terms of accuracy [4]. They are not necessarily more efficient; however, since our application is not significantly time-sensitive (i.e. the bus company can be informed of a snow-covered sidewalk within a few hours rather than within a few seconds), we prioritize accuracy over efficiency.

D. *Snow Detection.*

Most work has focused on detecting the presence of snowfall [5]–[8] and localizing the presence of vehicles and other objects in adverse weather conditions such as snow [9]–[11]. However, in addition to detecting snow, our application has the added complication of localizing the positions of sidewalks that are occluded by snow.

To our knowledge, there exists no dataset that contains labeled snow-covered sidewalks. Synthetic images are commonly used to artificially enlarge datasets; however, they are difficult to render realistic-looking [12]. Furthermore, training a deep learning model to classify an image as a "snow-covered sidewalk" would not be straightforward as any miscellaneous snow-covered area could look identical to a snow-covered sidewalk (Fig. 3).

Fig. 3: Classifying an image as a snow-covered sidewalk is difficult because the area under the snow is not visible.

E. *Image Localization.*

LiDAR is often used to localize the positions of objects surrounding an autonomous vehicle, such as other vehicles [13]–[18]. However, LiDAR is expensive, and we already have thousands of images available from bus cameras [19]. Furthermore, weather conditions like snow can interfere with LiDAR measurements [19].

Some methods have been developed for an analogous problem of localizing roads in adverse weather conditions. Some applications depend on a previously generated map of the terrain [20]; we apply a similar idea of generating a preconception of where the sidewalks should be. A few methods use the geometry of the road, such as the vanishing point of the road and the horizon in the image, to generate an expected target area for where the road could be [21], [22]. Another method uses self-supervision to generate a pseudo-mask of where the road is expected to be [23]. However, these approaches are more effective in weather conditions under which the road is partially occluded, such as fog or rain. They are generally unable to localize roads that are fully occluded by snow. Furthermore, these methods target the autonomous driving domain where they must anticipate completely novel surroundings on any given drive. On the other hand, we leverage the fact that we work with images from a mostly repetitive bus route.

Structure from Motion (SfM) [24], [25] is a classic computer vision algorithm that uses several two-dimensional images taken at different angles of a scene to construct a three-dimensional point cloud representation of the scene. Furthermore, SfM can deduce the pose of the camera for each image and for new images of the same scene [26]. We use a pipeline for SfM called COLMAP [27], [28]. More specifically, COLMAP implements incremental SfM which gradually adds images when reconstructing a scene (Fig. 4); this is as opposed to global SfM [29].

Visual odometry (VO) methods can also localize images [30] and tend to run faster than COLMAP; however, they are not as accurate. Furthermore, VO methods that involve deep learning [31] are inherently data-hungry, and our method aims to reduce the amount of annotated data needed. Since our application is not significantly time-sensitive, and we need to accurately classify a sidewalk as snow-covered or clear, we require a robust pipeline like COLMAP. Furthermore, the COLMAP software is well-documented and often referenced as a baseline method by these new methods.

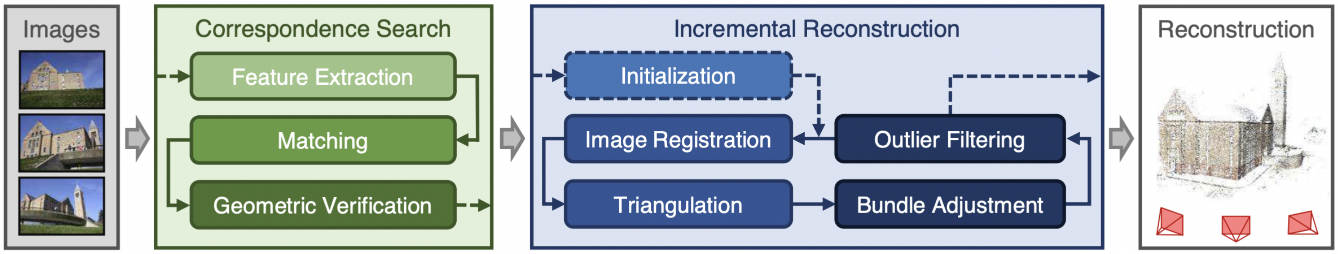


Fig. 4: Steps for incremental SfM. Figure from Fig. 2 in [27].

III. METHOD

For simplicity, we describe our method for one stretch of road that includes a bus stop.

A. *Data collection*

We use images captured from the dash camera on a bus (Fig. 5a) that travels around Pittsburgh and Washington County, Pennsylvania (Fig. 5b). The camera captures 5 frames per second. Duplicate and blurry images are removed, and the remaining images are sent to a server via the BusEdge Platform [1]. The images, along with their GPS and IMU information, are stored as EXIF files in folders of .bag files. We have data beginning from February 2021.

We choose images corresponding to a stretch of the route with a visible sidewalk in clear weather. Then, using the GPS information of the selected images, we filter images from other clear-weather days to obtain a few runs of the same sidewalk stretch.

We omit images within the selected GPS range where the bus travels on the opposite side of the road (i.e. returning on the same route) so as to focus on one side of the road. This omission is not strictly necessary. Some images have a strong glare from the sunlight and consequently the scenes in these images are extremely dim, so we remove these images. In the end, we keep three runs of the same stretch of sidewalk in clear weather. We use a similar process to collect different categories of sidewalk images for evaluation (described in Sec. IV).

B. *Reconstruction of ground truth sidewalks*

The steps detailed in this section are adapted from an analogous work in the detection of changes in crosswalks [32]. For this application, the overall idea is to save the sidewalk locations into a 3D rendering of the scene (Fig. 6a).

1) **Use reference images to render a point cloud of the scene.**
   We feed the images of the sidewalk stretch from multiple clear-weather runs (i.e. *reference* images) into COLMAP [27], [28] and obtain a point cloud representation of the stretch.
   SfM has three main steps [33] (Fig. 4):
   a) *Identify keypoints.*
      Keypoints are points in the scene that are somewhat salient and specific to the scene. Objects like monuments and store signs tend to provide robust keypoints.
   b) *Represent the keypoints as vectors.*
      COLMAP uses the SIFT descriptor [34] to extract the features of the keypoints and their local surroundings as vectors [1].
   c) *Reconstruction.*
      First, the pairwise relationships between images are determined by using RANSAC [36] and the extracted feature vectors. Then, reconstruction begins with the image pair containing the most inliers. Images are gradually added while solving bundle adjustment.

   We run COLMAP on images from multiple runs of the same stretch because one run tends to produce too sparse a point cloud. Since the bus camera is unlikely to be in the exact same orientation between runs (e.g. not always centered in the lane), we are effectively guaranteed reasonable stereo pairs which improves the 3D reconstruction.
2) **Estimate the ground plane of the scene.**
   First, we segment the road in each reference image by applying the Mask2Former [3] model. We select the

(a) Picture of the bus and camera. See [1] for camera specifications.

(b) Bus route.

Fig. 5: Information about the bus that is used to collect data.

[1]There also exist other descriptors such as SuperPoint [35] that can be used instead.

< TABLE OF CONTENTS

panoptic segmentation model with a Swin-L (IN21k) backbone that is pre-trained on Mapillary Vistas. Mapillary Vistas [37] is a dataset that contains street-level images, and its panoptic segmentation categories include snow, sidewalks, and roads.

Then, we use the pixel-to-3D point correspondences provided by COLMAP to identify the points in the point cloud that correspond to the road pixels. Next, we use RANSAC to fit a plane to the identified points; this plane is the estimated ground plane. Finally, we re-orient all the points such that the z-axis of the point cloud aligns with the normal of the ground plane.

3) **Segment the sidewalk in each reference image.**
For this step, we obtain masks of the sidewalks in the reference images by again applying the Mask2Former model [3]. Using the assumption that the bus is driving on the right side of the road as is the convention in the United States, we omit identified sidewalk pixels that are to the left of and/or above the midpoint of the image. This helps restrict the view of the bus to the sidewalk closest to it. The driving assumption would need to be adjusted in locations where the driving conventions differ.

4) **Use the estimated ground plane to project the sidewalk masks into the point cloud and save the projected points.**
In order to save the sidewalk locations into the point cloud, we use the road to determine the homography from the image to the point cloud. We can assume that the road is a flat reference area and that the slight lift of the sidewalk does not contribute much error as seen in our example qualitative results (Fig. 9).

Let us consider one reference image. First, we find the homography from the road pixels in the image to the corresponding 3D points that lie on the estimated ground plane. This is effectively the homography from the image to the estimated ground plane. Next, we use the homography matrix to project each pixel in the sidewalk mask onto the estimated ground plane. Lastly, we save coordinates of the projected sidewalk points. We repeat this process for each reference image, and the combined points form a 3D model of the expected sidewalk locations.

C. *Classification of query image*

In our application, *query* images are images from future runs of the bus when there could be snowfall. This part of the method involves classifying a query image as Clear or Snow-covered (Fig. 6c).

1) **Check if the query image belongs to the scene.**
For a given query image, we use GPS information to check if the query belongs to the point cloud. If the query is within the GPS range for the scene, we proceed.
2) **Identify the snow coverage in the query, if any is present.**
We use the Mask2Former model [3] to segment the snow in the query and proceed if snow is present.
3) **Estimate the camera pose of the query.**
If the query does belong to the scene, we add the query to the collection of reference images and re-run COLMAP to obtain an estimated camera pose for the query. Sometimes, COLMAP needs to be re-run more than once to obtain an accurate pose for the query. This reconstruction does not take as long as the initial point cloud rendering because the query is simply added to the existing reconstruction.
4) **Compare the snow coverage to the expected sidewalk area.**
Using the estimated camera pose and ground plane, we project the saved sidewalk points from the point cloud into the image. Finally, we calculate the proportion of the projected sidewalk that overlaps with the snow.
5) **Generate an alert if the snow significantly covers the expected sidewalk area.**
If the coverage is greater than a set threshold, we generate an alert. See Sec. IV for details about selecting an alert threshold.

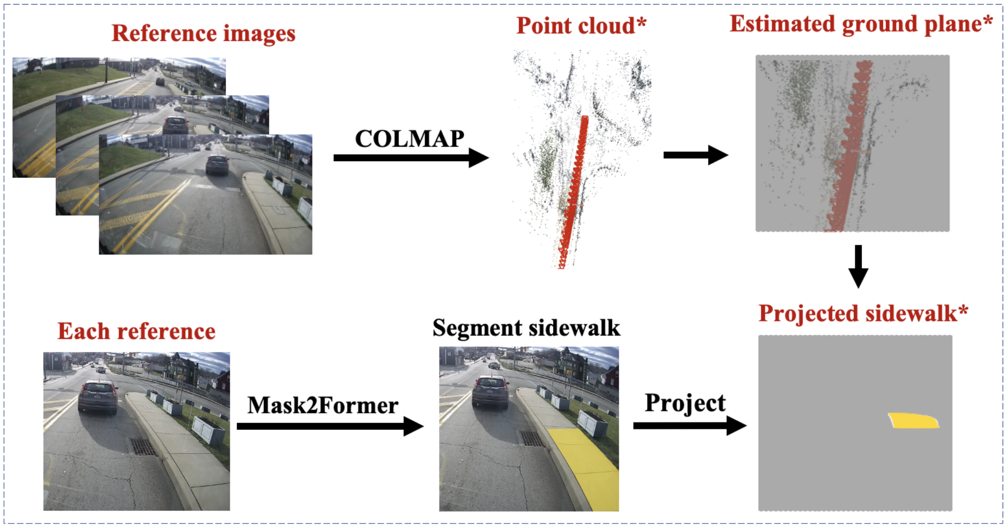
IV. RESULTS

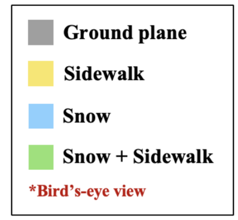
We evaluate by first reconstructing the ground truth sidewalks for two stretches that include bus stops. These stretches were chosen based on where there were images from the categories described below. We select and evenly split 66 test images taken during daylight hours into three categories:

(a) (b) (c)

Fig. 7: Categories of test images. **(a)** is clear, **(b)** is snow-covered, and **(c)** is cleared.

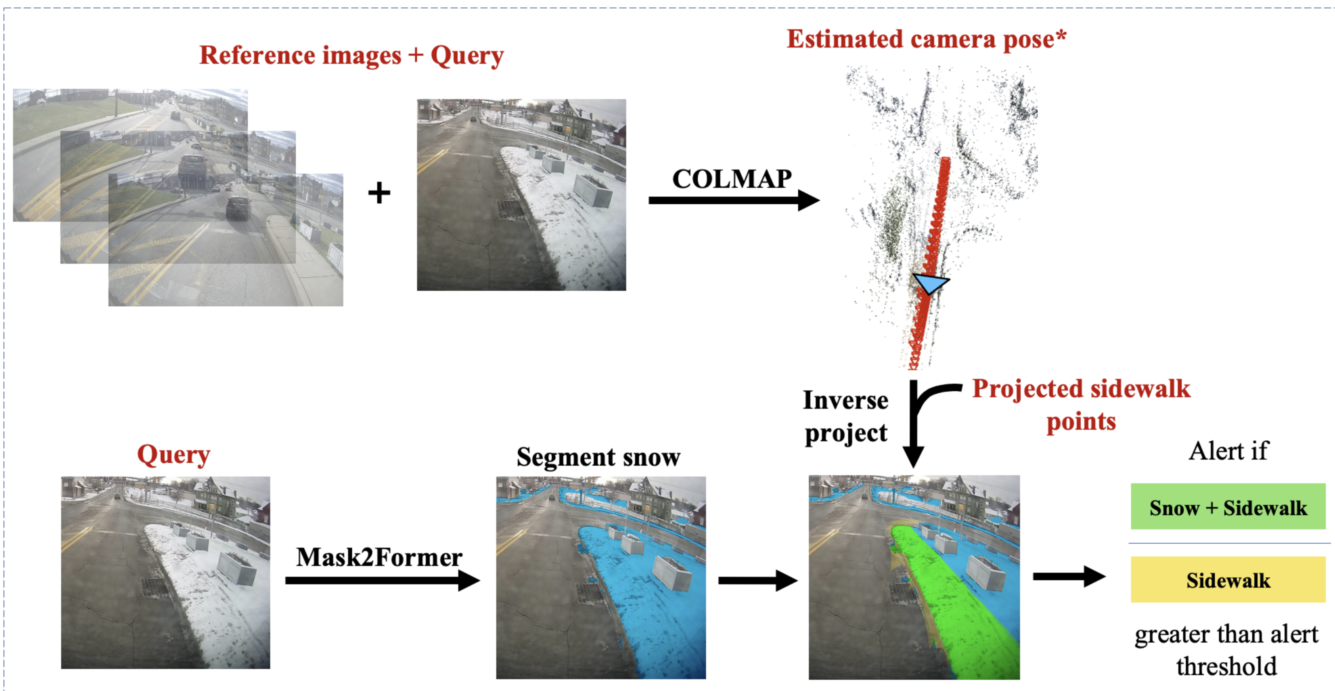
1) **Clear**
This category includes images in clear weather from February 2022 when the sidewalk in the chosen stretch is clear (Fig. 7a). These images should be classified as Clear and are distinct from the reference images originally used to render the point cloud.
2) **Snow-covered**
This category includes images in which the sidewalks are obscured by snow (Fig. 7b). These images should be classified as Snow-covered and were taken on January 7$^{\text{th}}$, 2022 when a snowstorm occurred in the Pittsburgh area.
3) **Cleared**
This category includes images in which the sidewalks are surrounded by but not covered with snow (i.e. the





(a) **Reconstructing the ground truth sidewalks.**

(b) **Key.**



(c) **Classifying a query image.**

Fig. 6: We describe this method in terms of one stretch of road that has a bus stop: Our method begins by using clear-weather images of the stretch to render a 3D model of the sidewalks as shown in **(a)**. This involves running SfM on the reference images to render a point cloud of the scene, estimating the ground plane of the point cloud, and then projecting the portion of the sidewalk mask that is closest to the bus in each reference image onto the estimated plane. When the bus encounters snow coverage, our method compares the snow to the expected sidewalk area as shown in **(c)**. This process involves estimating the camera pose of the query by re-running COLMAP with the query added to the reference images and using the estimated pose to project the saved sidewalk points into the query image (i.e. the inverse projection of **(a)**). Finally, if the proportion of the expected sidewalk area that is covered by snow exceeds a set alert threshold, we generate an alert.

sidewalks have been cleared) (Fig. 7c). These images are important because snow is present in the image but does not obstruct the sidewalk. These images should be classified as Clear; we do not need to distinguish between clear and cleared sidewalks. Like the snow-covered images, these images were taken on January 7$^{\text{th}}$, 2022 though in a different stretch.

For each category, we obtain the percent of images that are correctly classified as either Snow-covered or Clear across incremented alert thresholds. For clear images, our method trivially performs well across all thresholds because there is no snow present in any of the images (Fig. 8a). For snow-covered sidewalks, our method performs well until a threshold of around 0.7 (Fig. 8b). This trend is reasonable because a snow-covered sidewalk will have a high, but not necessarily perfect, overlap with the snow in the image. Finally, for cleared sidewalks, our method performs better as the threshold increases (Fig. 8c). This trend is reasonable because a stricter (i.e. higher) threshold will classify more images as Clear.

Since we need a threshold that will perform well across all categories, we identify 0.58 to 0.62 as a good range of thresholds. We include some example qualitative results from each test category at an alert threshold of 0.60 (Fig. 9). The saved sidewalk points that are projected into the query image generally align well with the real sidewalk in the query. The few misalignments that we observe (Fig. 9c) are most likely due to an inaccurate estimated camera pose for the query and/or for some of the reference images when projecting the ground truth sidewalk masks into the point cloud.

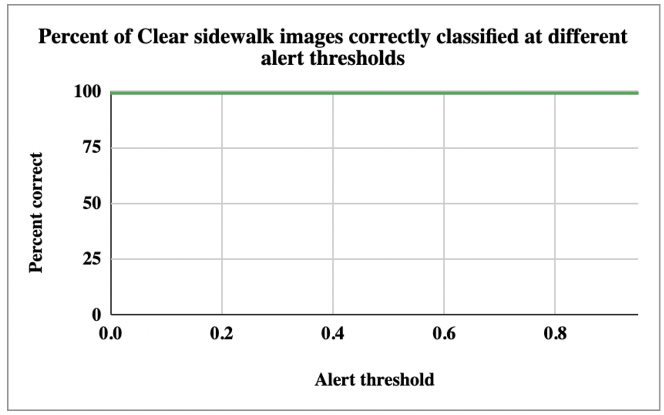
V. CONCLUSIONS

In this paper, we present and demonstrate a less data-intensive method for detecting snow-covered sidewalks along bus routes. Our method leverages Structure from Motion to learn the expected locations of sidewalks during clear weather and then uses the learned locations to determine if the sidewalks become covered with snow. By evaluating on different categories of sidewalks, we identify a range of thresholds across which this method performs well for our bus route.

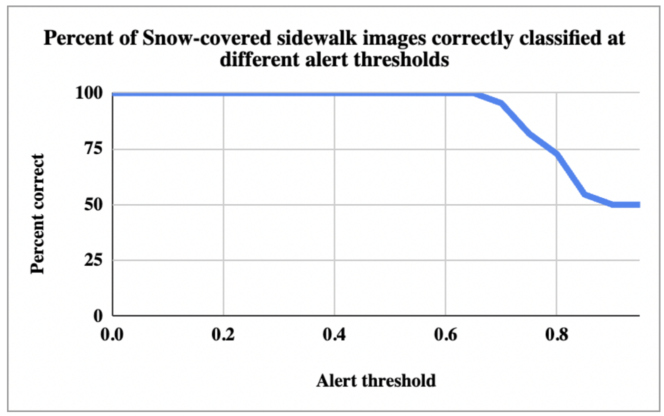
For our particular application, an immediate extension of this method is to incorporate GPS information to form a full route of point clouds. However, our method can also extend to other hazardous conditions such as snow-covered bike lanes or roads.

One limitation of this method is its dependence on keypoints. The effectiveness of the SIFT descriptors in the SfM process depends on the presence of robust and unique keypoints. This, in turn, can affect the estimated camera poses for each image.

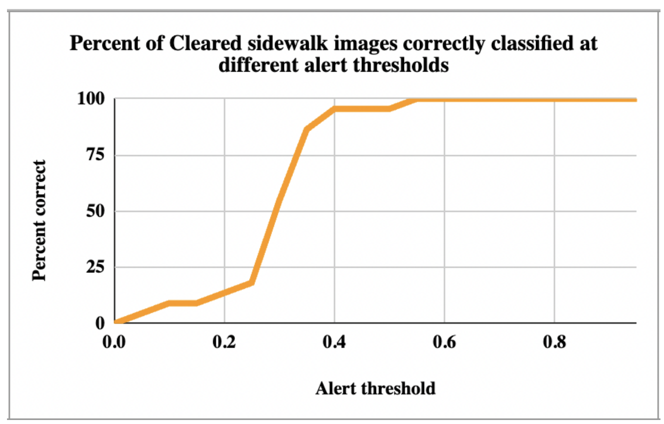
In urban scenes, there exist many buildings, signs, and sometimes monuments that provide such keypoints. However, there may not exist many salient keypoints in rural areas Likewise, in night settings, keypoints can be less visible. In these cases, it would be interesting to experiment with adding GPS information for feature matching in SfM.



(a) **Results for clear sidewalks.** Our method trivially performs well at all thresholds because there is no snow in images from this category.



(b) **Results for snow-covered sidewalks.** Our method performs well until around a threshold of 0.70 when the percent of images classified correctly begins decreasing more rapidly.



(c) **Results for cleared sidewalks.** Our method performs well after a threshold of 0.40 when the percent of images classified correctly begins to stabilize.

Fig. 8: These graphs depict the percent of images correctly classified across alert thresholds from 0 to 0.95 incremented by 0.05. **(a)** shows results for clear sidewalks, **(b)** shows results for snow-covered sidewalks, and **(c)** shows results for cleared sidewalks. Alert thresholds ranging from 0.58 to 0.62 produce an optimal performance across all three test categories.

(a) This is an example of a clear query that is correctly classified as Clear (coverage = 0). No snow is identified in the image, and the projected sidewalk (shown in yellow) aligns fairly well with the actual sidewalk.

(b) This is an example of a snow-covered query that is correctly classified as Snow-covered (coverage = 0.91). The projected sidewalk (mostly colored in green due to overlap with snow) aligns almost perfectly with the actual sidewalk which is not visible.

(c) This is an example of a cleared query that is correctly classified as Clear (coverage = 0.36). The projected sidewalk aligns, though not perfectly, with the sidewalk in the image. The misalignment present towards the top left can occur due to an inaccurate estimated camera pose for the query or for some of the reference images when reconstructing the ground truth sidewalks. In this case, the latter probably occurred as there are full patches of projected sidewalk off to the side.

Fig. 9: These images depict an example of a qualitative result for each of the three test categories at an alert threshold of 0.60. The left panels depict a query image from each category. The right panels display snow in blue, the saved sidewalk projected into the image in yellow, and overlap between snow and the projected sidewalk in green.

ACKNOWLEDGMENT

This research was supported by the National Science Foundation under Award No. 2038612. Data and background software were provided by projects sponsored by Carnegie Mellon University's Mobility21 National University Transportation Center, which is sponsored by the United States Department of Transportation. We would also like to thank Dr. John M. Dolan, Rachel Burcin, and the RISS program and sponsors for further supporting this project.

REFERENCES

[1] C. Ye, "BusEdge: Efficient Live Video Analytics for Transit Buses via Edge Computing," Master's thesis, Carnegie Mellon University, Pittsburgh, PA, July 2021.

[2] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, "Panoptic Segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[3] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention Mask Transformer for Universal Image Segmentation," 2022.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[5] Q. A. Al-Haija, M. Gharaibeh, and A. Odeh, "Detection in Adverse Weather Conditions for Autonomous Vehicles via Deep Learning," *AI*, vol. 3, no. 2, pp. 303–317, 2022. [Online]. Available: https://www.mdpi.com/2673-2688/3/2/19

[6] M. N. Khan and M. M. Ahmed, "Snow Detection using In-Vehicle Video Camera with Texture-Based Image Features Utilizing K-Nearest Neighbor, Support Vector Machine, and Random Forest," *Transportation Research Record*, vol. 2673, no. 8, pp. 221–232, 2019. [Online]. Available: https://doi.org/10.1177/0361198119842105

[7] J. Carrillo, M. Crowley, G. Pan, and L. Fu, "Comparison of Deep Learning models for Determining Road Surface Condition from Roadside Camera Images and Weather Data," in *The Transportation Association of Canada and Intelligent Transportation Systems Canada Joint Conference (TAC-ITS)*, Halifax, Canada, 2019, p. 16.

[8] A. Lagorio, E. Grosso, and M. Tistarelli, "Automatic Detection of Adverse Weather Conditions in Traffic Scenes," in *2008 IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance*, 2008, pp. 273–279.

[9] X.-Z. Chen, C.-M. Chang, C.-W. Yu, and Y.-L. Chen, "A Real-Time Vehicle Detection System under Various Bad Weather Conditions Based on a Deep Learning Model without Retraining," *Sensors*, vol. 20, no. 20, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/20/5731

[10] M. Hnewa and H. Radha, "Object Detection Under Rainy Conditions for Autonomous Vehicles: A Review of State-of-the-Art and Emerging Techniques," *IEEE Signal Processing Magazine*, vol. 38, no. 1, pp. 53–67, 2021.

[11] M. Hassaballah, M. A. Kenk, K. Muhammad, and S. Minaee, "Vehicle Detection and Tracking in Adverse Weather Using a Deep Learning Framework," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4230–4242, 2021.

[12] A. Kar, A. Prakash, M.-Y. Liu, E. Cameracci, J. Yuan, M. Rusiniak, D. Acuna, A. Torralba, and S. Fidler, "Meta-Sim: Learning to Generate Synthetic Datasets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[13] W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song, "L3-Net: Towards Learning Based LiDAR Localization for Autonomous Driving," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 6382–6391.

[14] J. Levinson, M. Montemerlo, and S. Thrun, "Map-Based Precision Vehicle Localization in Urban Environments," in *Robotics: Science and Systems*, 2007.

[15] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," *2010 IEEE International Conference on Robotics and Automation*, pp. 4372–4378, 2010.

[16] R. Kummerle, D. Hahnel, D. Dolgov, S. Thrun, and W. Burgard, "Autonomous driving in a multi-level parking structure," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3395–3400.

[17] R. W. Wolcott and R. M. Eustice, "Fast LIDAR localization using multiresolution Gaussian mixture maps," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2814–2821.

[18] X. Chen, I. Vizzo, T. Läbe, J. Behley, and C. Stachniss, "Range Image-based LiDAR Localization for Autonomous Vehicles," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5802–5808, 2021.

[19] P. Wang, "Research on Comparison of LiDAR and Camera in Autonomous Driving," *Journal of Physics: Conference Series*, vol. 2093, no. 1, p. 012032, nov 2021. [Online]. Available: https://doi.org/10.1088/1742-6596/2093/1/012032

[20] R. Firoozi, J. Guanetti, R. Horowitz, and F. Borrelli, "Vehicle Localization and Control on Roads with Prior Grade Map," in *2018 IEEE Conference on Decision and Control (CDC)*, 2018, pp. 6982–6987.

[21] E. J. Almazan, Y. Qian, and J. H. Elder, "Road segmentation for classification of road weather conditions," in *Computer Vision – ECCV*

2016 Workshops, G. Hua and H. Jégou, Eds. Cham: Springer International Publishing, 2016, pp. 96–108.

[22] G. Cheng, Y. Qian, and J. H. Elder, "Fusing Geometry and Appearance for Road Segmentation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[23] D. Kothandaraman, R. Chandra, and D. Manocha, "SS-SFDA: Self-Supervised Source-Free Domain Adaptation for Road Segmentation in Hazardous Environments," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, October 2021, pp. 3049–3059.

[24] S. Ullman, "The Interpretation of Structure from Motion," *Proceedings of the Royal Society of London. Series B, Biological Sciences*, vol. 203, no. 1153, pp. 405–426, 1979. [Online]. Available: http://www.jstor.org/stable/77505

[25] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[26] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo Tourism: Exploring Photo Collections in 3D," *ACM Trans. Graph.*, vol. 25, no. 3, p. 835–846, jul 2006. [Online]. Available: https://doi.org/10.1145/1141911.1141964

[27] J. L. Schönberger and J.-M. Frahm, "Structure-from-Motion Revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[28] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise View Selection for Unstructured Multi-View Stereo," in *European Conference on Computer Vision (ECCV)*, 2016.

[29] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher, "Discrete-Continuous Optimization for Large-scale Structure from Motion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[30] C. Campos, R. Elvira, J. J. G. Rodr'iguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, pp. 1874–1890, 2021.

[31] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2043–2050.

[32] T. Bu, "Towards HD Map Updates with Crosswalk Change Detection from Vehicle-Mounted Cameras," Master's thesis, Carnegie Mellon University, Pittsburgh, PA, August 2022.

[33] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, "A Survey of Structure from Motion," *Acta Numerica*, vol. 26, p. 305–364, 2017.

[34] G. LoweDavid, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, 2004.

[35] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[36] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Commun. ACM*, vol. 24, no. 6, p. 381–395, jun 1981. [Online]. Available: https://doi.org/10.1145/358669.358692

[37] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder, "The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 5000–5009.

[38] Stanford Artificial Intelligence Laboratory et al., "Robot Operating System." [Online]. Available: https://www.ros.org/

[39] B. Cheng, A. G. Schwing, and A. Kirillov, "Per-Pixel Classification is Not All You Need for Semantic Segmentation," 2021.

Sim2Real transfer for capacitive sensors utilizing Assistive Gym's capacitive sensing simulation framework

Jacob M. Delgado-López[1], Yufei Wang[2] and Zackory Erickson[3]

***Abstract*— Capacitive sensing is a type of near-range sensing technology with a unique feature of sensing through non-conductive occlusions. Such a feature is especially useful for assistive robots that provide caregiving services as they can improve the quality of life of people with disabilities. For example, in robot-assisted dressing, capacitive sensors can be used to track the person's arms even under the occlusion of the cloth. That being said, directly designing and collecting data with capacitive sensors for robots interacting with humans in the real world can be slow, costy, and unsafe. On the other hand, robotics simulation provides a cheaper, safer, and more instructive alternative to real-world experimentation. In this project, we aim 1) to leverage a recently developed capacitive sensing simulation framework in Assistive Gym to optimize the design of capacitive sensors for assistive robots, 2) to learn a controller using simulated capacitive data for several assistive tasks, and 3) perform simulation to real-world (Sim2Real) transfer of the results to real-world robots. We first show that the gap between the simulation and the real world can be closed via optimizing simulation parameters. We then optimize the capacitive sensor design and train robotic controllers for a set of caregiving tasks in Assistive Gym using a large amount of the simulated capacitive data. At last, we build real-world replications of the simulated assistive tasks and show the capacitive sensor design and controllers obtained in the simulation can be transferred to real-world robots. Overall, we showcase the benefits of utilizing capacitive sensors in caregiving tasks and the advantages of utilizing simulation to train capacitive sensing models prior to real world experimentation.**

***Index Terms*— Physical Human-Robot Interaction, Physically Assistive Devices, Simulation**

I. INTRODUCTION

During COVID-19, 18.4% of older adults living alone reported difficulties with activities of daily living (ADLs) which ranged from 8.8% in Switzerland up to 29.2% in the USA [1]. Furthermore, only 56.8% of those reporting difficulties received ADL assistance [1]. Robotic assistance presents an alternative method for providing help to those who require support conducting everyday tasks. With this in mind, capacitive sensing, a novel near-range sensing technology, has been utilized for caregiving services as it provides the unique advantage of sensing through non-conductive oclusions. For example, in robot-assistive bathing, capacitive sensors can be used to track the position of a person's limb despite the oclusion of the cloth being used for bathing. However, directly designing and collecting data with capacitive sensors for robots interacting with humans in the real world can be slow, costy, and unsafe.

To overcome this, robotics simulation is used as it provides a cheaper, safer and more instructive alternative to real-world experimentation. In this work, we aim to leverage the novel capacitive sensing simulation framework CapSense in Assistive Gym to optimize the design of capacitive sensors for use in assistive robots, to learn a controller using simulated capacitive data for several assistive tasks, and perform simulation to real-world (Sim2Real) transfer of the results to real-world robots.

Our approach looks to capture real-world capacitive sensing data around a human limb with the use of a Stretch RE1 robot with a Dexwrist. The robot utilizes a 3D printed tool with a mounted capacitive sensor and follow predetermined trajectories around a static human limb which allows us to capture the changes in capacitance as the sensors approach the limb. Hence, we can map the changes in capacitance and the distance from the sensors to the closest point on the arm; allowing us to predict the sensor's position based on the capacitance reading. Then, the same trajectories are replicated in simulation where we match our simulated data with our real world data via optimization to bridge the sim2real gap. With this, we can train a controller in simulation for assistive tasks and transfer this into a real world environment.

We first show that the gap between the simulation and the real world can be closed via optimizing simulation parameters. With this, several assistive tasks can be performed in simulation with high accuracy to their real world counterparts. Hence, we can train a policy in simulation utilizing the capacitance measurements collected. Lastly, we transfer this controller into the real world and investigate its performance which will allow us to optimize the design of the capacitive sensors.

II. RELATED WORKS

A. Capacitive Sensors

Capacitive sensing technology has been implemented in common consumer goods such as touchscreens, MP3 players and mobile phones [2] [3]. More recently, capacitive sensing technology can be seen in industrial applications such as proximity sensing, position sensing, humidity sensing and tilt sensing [4]. Mayank Shadwani et al. [5] provides an overview of how capcitive sensing is being utilized in modern

[1]Jacob Delgado is a Robotics Institute Summer Scholar at Carnegie Mellon University and an undergraduate junior in Computer Science and Engineering, University of Puerto Rico, Mayagüez, 00682, Puerto Rico `jacob.delgado@upr.edu`

[3]Yufei Wang is a PhD student with the RCHI lab at the Robotics Institute in Carnegie Mellon University, PA 15213, USA `yufeiw2@andrew.cmu.edu`

[2]Zackory Erickson is an Associate Professor at the Robotics Institute in Carnegie Mellon University and the Principal Investigator at RCHI lab, PA 15213, USA `zackory@cmu.edu`

technology, especially in human interaction applications such as helmets that utilize the human ears' capacitance as input to minimize the percentage of road accidents as the vehicle will only move as long as the helmet is being worn by the driver.

In the area of robotics, capacitive sensors have frequently been used for proximity sensing in cluttered environments [6]. Alwin Hoffman et al. [6] present a comprehensive study on proximity sensing in a human-centered workspace. They record an environment model containing the expected sensor values for relevant robot poses which leads to accurate distance estimation and thus real-time reaction to these distances regardless of the presence of other conductive objects in the workspace. Wearable capacitive sensors have also gained prominence with an array of applications such as health monitoring [7] [8], locomotion classification [9] and activity recognition [10].

Capacitive sensors have also been used in Physical human-robot interaction as the sensors allow the robots to accurately sense the human body, follow trajectories around the body, and track human motion [11]. With this in mind, capacitive sensors have gained prominence for carrying out caregiving tasks as they are mounted on the robot's end effector and provide high resolution force sensing when interacting with human subjects. For example, Zackory Erickson et al. [12]. have utilized capacitive sensors for tracking the human pose during assistive dressing. By tracking the human's change in pose in real-time, a robot can adjust for errors in the estimated pose of a person and physically follow the contours and movements of the person while providing dressing assistance [12]. Moreover, multidimensional capacitive sensing have also been utilized for assistive bathing as it allows the robot to follow the contours of the human body during the bathing task [13].

B. Sim2Real

Simulation is highly prevalent in the field of research as it provides a cheaper, safer, and more informative alternative to real-life experimentation [14]. Simulation has a variety of applications from healthcare applications [15] to crop production [16]. In the field of robotics, simulation is a powerful visualization, planning, and strategic tool as it allows experimentation before real-world implementation. However, the current state of simulation in robotics is plagued with various challenges. Afsoon Afzal et al. [17] provide a comprehensive study on how researchers in the field of robotics detail their struggles, the gap between simulation and reality, a lack of reproducibility, and considerable resource costs associated with using simulators.

Sim2Real transfer in robotics is the concept of taking a controller learned in simulation and transfer the results learned with simulated data to a real-world robot. To successfully complete the transfer, the gap between a simulated experiment and a real world experiment must be negligible. Thus, proving that the simulated controller has practical use as it obtains similar results to the real world environment. Sim2Real transfer is utilized with various techniques in mind. Manuel Kaspar et al. [18] present Sim2Real transfer of a Reinforcement Learning controller without implementing Dynamics Randomization which speeds up training, can increase performance and reduces the number of hyperparameters.

C. Robotic Assistance

As robots have become more accepted in the area of robot-human interaction [19], several works have explored robot-assistance for a variety of caregiving tasks. For example, robotic arms have been used for robotic-assistive feeding [20]. Robotic arms have also been used to assist with ADL's of people living with limited mobility or dexterity. Nathaniel Dennier et al. [21] present an approach for enabling general robot manipulators to assist with a hair-combing task.

Researchers have also investigated the use of robots in assistance with bathing tasks, M. Manti et al. [22] implement soft robotics to build soft modular manipulators for assistive robotics that can safely interact with people during a bathing task. Moreover, there has been multiple efforts to incorporate capacitive sensing into robotic-bathing assistance [23], giving the robot the advantage of sensing the human body through occlusions. We build upon this prior research by replicating similar tasks in simulation and tweaking parameters in order to obtain a faithful reproduction of the data obtained in a real world environment.

III. Experiments

A. Capacitive Sensor Design

First of all, a tool was designed and 3-D printed with the purpose of performing a cleaning task on a person's limb while taking into account safety and efficiency. Moreover, the tool looked to accommodate a mounted capacitive sensor created with the use of a teensy-3.2 development board and copper foil tape. The teensy's unique touch pin sensor hardware was utilized to implement a sensor design that included a 3x2 grid with 6 electrodes.

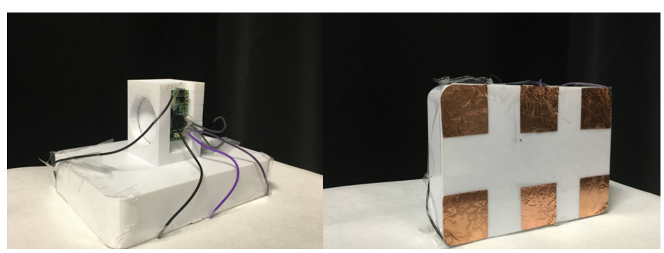


Fig. 1. Capacitive Sensor mounted on 3-D printed washing tool

B. Real World Data collection

Data collection is carried out by utilizing a Stretch RE1 robot with a Dextwrist as it allows us great range of motion. A set of seven trajectories are determined for the data collection. Amongst these trajectories, three of them focus on translational motions, two focus on rotational motions using the Dextwrist, and the last two are combinations of both translational and rotational motions. For example, for one of the translation motions, the trajectory is to first lift the stretch arm 5 cm above the human arm and then move the sensor straight across the arm. On the other hand, for the rotational motions, one of the trajectories is to lift up the stretch arm 5 cm above the human arm and then move the sensor in a semicircular fashion. These tasks are all performed on a static human arm with a circumference of approximately 2.30 m.

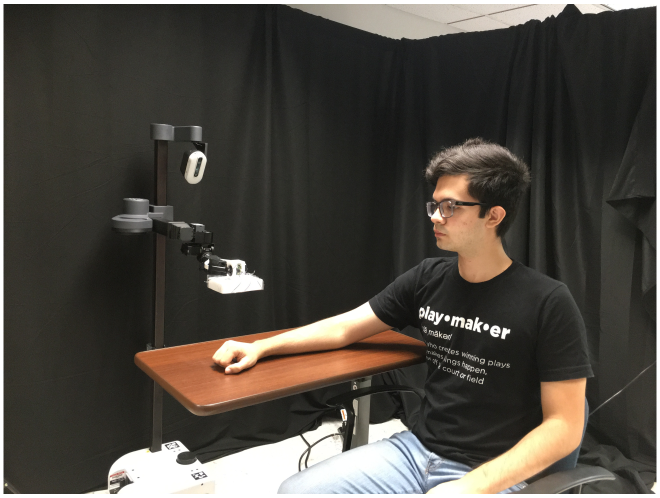


Fig. 2. Experimental setup for collecting data of different trajectories around the human arm

With this setup, we capture the changes in capacitance as the sensors approaches the arm, and also record the distance between the sensor and the human arm. This allows us to learn a model that maps the changes in capacitance to the distance from the sensors to the closest point on the arm, which allows us to predict the sensor's position based on the capacitance reading. The data collected from the varying trajectories show the expected behaviour as electrodes 1-3 (front of the tool) have a spike in capacitance first while 4-6 present the same behaviour a few seconds later. This occurs as electrodes 1-3 pass by the human arm before the latter.

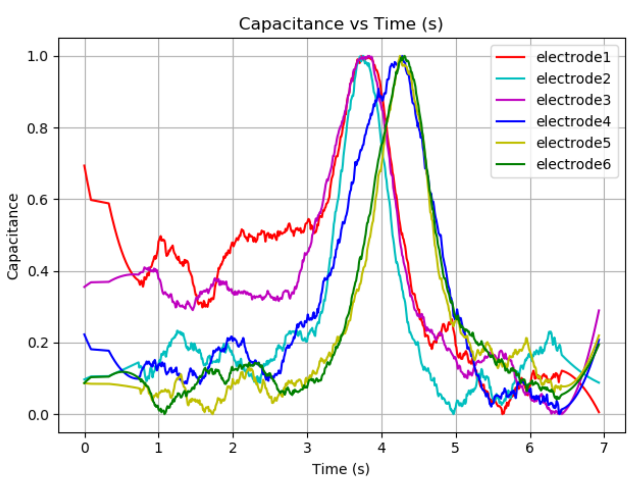


Fig. 3. Capacitance vs, Time plot of the sensor moving in a straight fashion through the arm

C. *Replication in Simulation*

After performing the trajectories and collecting data in the real world, the same trajectories were replicated in simulation where we perform system identification in order to create a faithful simulation environment. Here, we look to capture capacitance readings and compare them to our real world dataset in order to optimize the simulation parameters via Covariance Matrix Adaptation Evolution Strategy (CMA-ES).

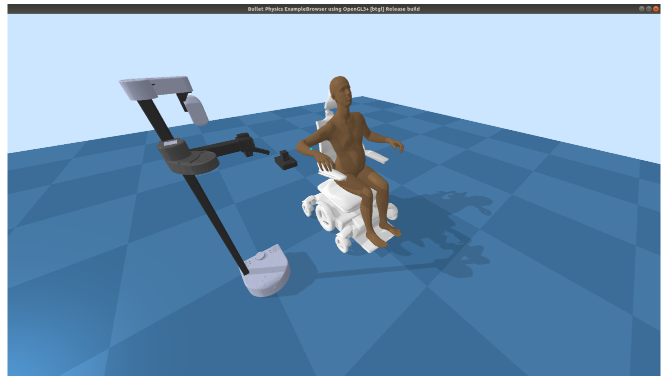


Fig. 4. Assistive Gym simulation environment setup for collecting data

IV. Future Work

For future work, we will further optimize the simulation parameters to obtain further fidelity in simulation as well as further reducing noise in real world data collection. In addition, we will look to utilize our capacitance readings and distance measurements as the basis for a pose estimation model. This way we can predict the pose of the human relative to the tool. At last, with the optimized simulation parameters that minimize the sim2real gap, we look to utilize the tuned simulation to collect data and train controllers and transfer these into a real word robot. Amongst the controllers, we will look into utilizing the washing tool to clean different human limbs such as arms, legs, chest, etc.

V. CONCLUSIONS

In summation, this work presents the foundation and initial findings for performing Sim2Real transfer of capacitive sensors utilizing Assistive Gym's capacitive sensing simulation framework. Additionally, we present some insight on the process for bridging the gap between simulation and the real world. Lastly, for future work, the simulation with optimized parameters will be used to train a robot controller for performing an assistive task utilizing the washing tool such as cleaning along different human limbs.

ACKNOWLEDGMENT

This work would not have been possible without the support of the RCHI lab and the Robotics Institute Summer Scholars Program. A sincere thank you to Dr. Zackory Erickson and Yufei Wang for their unconditional support and guidance throughout the summer. A large thank you to the RCHI lab members for their mentor-ship and assistance throughout the program. I would also like to thank Rachel Burcin and John Dolan for organizing RISS and making this summer experience possible. Lastly, thank you to the CMU Robotics Institute for funding this research.

References

[1] S. Chen, L. A. Jones, S. Jiang, H. Jin, D. Dong, X. Chen, D. Wang, Y. Zhang, L. Xiang, A. Zhu, and R. N. Cardinal, "Difficulty and help with activities of daily living among older adults living alone during the COVID-19 pandemic: a multi-country population-based study," *BMC Geriatrics*, vol. 22, no. 1, p. 181, Mar. 2022. [Online]. Available: https://doi.org/10.1186/s12877-022-02799-w

[2] D. Fischer, "Capacitive Touch Sensors," p. 12.

[3] B. Osoinach, "Proximity Capacitive Sensor Technology for Touch Sensing Applications," p. 12.

[4] E. Terzic, J. Terzic, R. Nagarajah, and M. Alamgir, "Capacitive Sensing Technology," pp. 11–37, 2012. [Online]. Available: https://doi.org/10.1007/978-1-4471-4060-3_2

[5] M. Shadwani, S. Sachan, and P. Sachan, "Capacitive Sensing & Its Applications," 2018, publisher: 6278. [Online]. Available: http://rgdoi.net/10.13140/RG.2.2.15752.32001

[6] A. Hoffmann, A. Poeppel, A. Schierl, and W. Reif, "Environment-aware proximity detection with capacitive sensors for human-robot-interaction," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 145–150.

[7] S. K. Kundu, S. Kumagai, and M. Sasaki, "A wearable capacitive sensor for monitoring human respiratory rate," *Japanese Journal of Applied Physics*, vol. 52, no. 4S, p. 04CL05, apr 2013. [Online]. Available: https://doi.org/10.7567/jjap.52.04cl05

[8] Bijender and A. Kumar, "Flexible and wearable capacitive pressure sensor for blood pressure monitoring," *Sensing and Bio-Sensing Research*, vol. 33, p. 100434, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214180421000398

[9] B. Chen, E. Zheng, X. Fan, T. Liang, Q. Wang, K. Wei, and L. Wang, "Locomotion mode classification using a wearable capacitive sensing system," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 21, no. 5, pp. 744–755, 2013.

[10] J. Cheng, O. Amft, G. Bahle, and P. Lukowicz, "Designing Sensitive Wearable Capacitive Sensors for Activity Recognition," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3935–3947, Oct. 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6507545/

[11] Z. M. Erickson, H. M. Clever, V. Gangaram, E. Xing, G. Turk, C. K. Liu, and C. C. Kemp, "Characterizing multidimensional capacitive servoing for physical human-robot interaction," *ArXiv*, vol. abs/2105.11582, 2021.

[12] Z. Erickson, M. Collier, A. Kapusta, and C. C. Kemp, "Tracking Human Pose During Robot-Assisted Dressing Using Single-Axis Capacitive Proximity Sensing," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2245–2252, July 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8307487/

[13] Z. Erickson, H. M. Clever, V. Gangaram, G. Turk, C. K. Liu, and C. C. Kemp, "Multidimensional capacitive sensing for robot-assisted dressing and bathing," in *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, 2019, pp. 224–231.

[14] R. G. Ingalls, "Introduction to simulation," in *Proceedings of the 2011 Winter Simulation Conference (WSC)*, 2011, pp. 1374–1388.

[15] K. Katsaliaki and N. Mustafee, "Applications of simulation within the healthcare context," *Journal of the Operational Research Society*, vol. 62, no. 8, pp. 1431–1451, 2011, pMID: 32226177. [Online]. Available: https://doi.org/10.1057/jors.2010.20

[16] G. Hoogenboom, "Contribution of agrometeorology to the simulation of crop production and its applications," *Agricultural and Forest Meteorology*, vol. 103, no. 1, pp. 137–157, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168192300001088

[17] A. Afzal, D. S. Katz, C. L. Goues, and C. S. Timperley, "A Study on the Challenges of Using Robotics Simulators for Testing," Apr. 2020, arXiv:2004.07368 [cs]. [Online]. Available: http://arxiv.org/abs/2004.07368

[18] M. Kaspar, J. D. Muñoz Osorio, and J. Bock, "Sim2real transfer for reinforcement learning without dynamics randomization," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4383–4388.

[19] A. K. Hall, U. Backonja, I. Painter, M. Cakmak, M. Sung, T. Lau, H. J. Thompson, and G. Demiris, "Acceptance and perceived usefulness of robots to assist with activities of daily living and healthcare tasks," *Assistive Technology*, vol. 31, no. 3, pp. 133–140, 2019, pMID: 29125804. [Online]. Available: https://doi.org/10.1080/10400435.2017.1396565

[20] G. Canal, G. Alenyà, and C. Torras, "Personalization framework for adaptive robotic feeding assistance," in *Social Robotics*, A. Agah, J.-J. Cabibihan, A. M. Howard, M. A. Salichs, and H. He, Eds. Cham: Springer International Publishing, 2016, pp. 22–31.

[21] N. Dennler, E. Shin, M. Matarić, and S. Nikolaidis, "Design and Evaluation of a Hair Combing System Using a General-Purpose Robotic Arm," Aug. 2021, arXiv:2108.01233 [cs]. [Online]. Available: http://arxiv.org/abs/2108.01233

[22] M. Manti, A. Pratesi, E. Falotico, M. Cianchetti, and C. Laschi, "Soft assistive robot for personal care of elderly people," in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, 2016, pp. 833–838.

[23] Z. Erickson, H. M. Clever, V. Gangaram, G. Turk, C. K. Liu, and C. C. Kemp, "Multidimensional capacitive sensing for robot-assisted dressing and bathing," in *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*. IEEE Press, 2019, p. 224–231. [Online]. Available: https://doi.org/10.1109/ICORR.2019.8779542

Shape and Texture Classification with ReSkin

Jacob Adkins[1], Raunaq Bhirangi[2], Abhinav Gupta2

***Abstract*— Tactile sensors have growing interest in robotics due to the desire for robots to perform dexterous manipulation tasks which vision alone cannot solve due to occlusion and other limitations. Examples of tasks are reaching into a bag and pulling out a specific object without being able to look into the bag. Or knowing how much force to apply when grasping. Recent developments in soft sensors have enabled passive conformal contact and active data collection. ReSkin is a recent soft sensor which enables low-cost magnetic sensing data from which inferences about contact can be made. This work utilizes a machine learning approach to utilize ReSkin magnetic data in such a way that robot systems can infer shape and texture information about objects that they touch. The ability to infer information from touch will enable robots to solve tasks that would otherwise be impossible with vision alone. This work coducts a classification study of the ReSkin sensor's ability to classify object shape and texture.**

***Index Terms*— Tactile Sensing, Object classification**

I. INTRODUCTION

The majority of robots that operate today use parallel jaw grippers. These types of robots are sufficient in settings such as factories which are controlled and have limited variability in the environment.

However, there has been a recent push in the field of robotics to design robots that can provide assistance to humans in unstructured environments, such as homes or offices.

Dexterous multi-fingered hands are essential for complex tasks such as sewing, typing, painting etc. Therefore, it seems that in order for robots to become more useful in real-world unstructured environments, they must become able control objects with dexterity; i.e., have complete 6 degree of freedom control over the object. In order to develop dexterous manipulation systems, we can look to humans for inspiration. It is apparent that human hands have all-over sensing tactile sense which is imperative to our ability to effectively interact with objects. This points to the need to integrate the tactile sensing modality into the system and the learning algorithm for manipulation.

As a first step, we look to ReSkin, a recently developed promising new tactile sensor which is conformal, replaceable, and allows for large area sensing [1]. ReSkin sensors detect changes in the magnetic field when contact is made with an object. Previously, [1] demonstrated that ReSkin data can be used to train a multi-layered perceptron model to predict position and force on the ReSkin sensor. In this work, we seek to demonstrate that ReSkin data can be used not just for detection force and position but for classifiying obejcts as well. In search of creating systems that allow for dexterous manipulation, several sensing modalities have been used; namely, vision and audio [2], [3]. However, one modality has been largely ignored, tactile sense. Tactile receptors are all over the skin of humans and numerous other animals. We use our sense of touch countless times per day and it seems intuitively obvious that tactile sensing is essential to performing daily tasks. Why then should robots not be able to feel objects just as we humans do. The combination of tactile sense with other modalities already in use should allow for the creation of more robust robot manipulation schemes.

II. Related Works

There has been recent success with several vision-based tactile sensors that have been proposed in recent years. Such as Gelsight [4] and DIGIT [5]. In addition, there is OmniTact which uses micro cameras to detect deformations of a gel-based skin [6]. These sensors have been popular due to their high resolution as well as the existence of neural architectures for processing these signals. These sensors however are bulky, difficult to fabricate and have limited spatial coverage. A number of other modalities like capacitative [7], resistive [8] and piezoelectric [9] sensing have also been explored as tactile sensing alternatives for robotics. These technologies however, often need direct electrical connections between the circuitry and the interface, lack shear sensing capabilities, and/or are difficult to scale-up in size.

In this paper, we examine ReSkin [1], a magnetic elastomer-based tactile sensor that seeks to avoid these pitfalls by using magnetic microparticles embedded in elastomer as the sensing interface and a circuit of magnetometers underneath to detect the deformation.

III. ReSkin Overview

The choice of using ReSkin was because of several attractive attributes that it provides. First, ReSkin is conformal, i.e. it conforms around objects that the robot is trying to grasp and allows for stable grasp. Secondly, ReSkin allows for large area sensing. Therefore, the robot is able to feel several sides of the object at once. And Finally, ReSkin is low cost as it only requires approximately $6 of materials to create a ReSkin sensor. Because of all of these reasons, ReSkin was chosen as the sensor to perform the object classification study with. ReSkin is composed of an

[1] Jacob Adkins is with the departments of mathematics and Computer Science, New College of Florida, Sarasota, FL, USA, jacob.adkins18@ncf.edu

[2]Raunaq Bhirangi and Abhinav Gupta are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, {rbhirangi,agupta} @andrew.cmu.edu

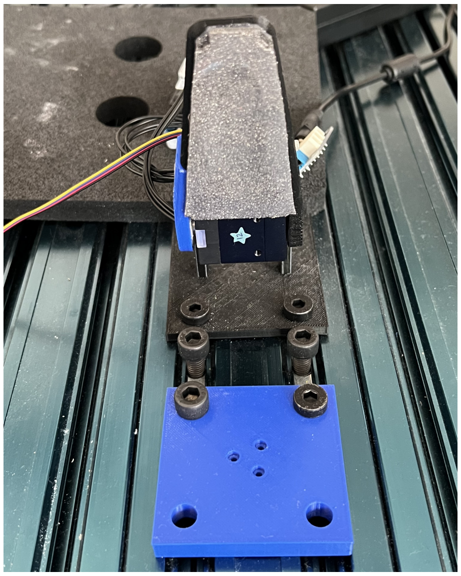


Fig. 1. 3D printed finger used in shape classification experiment

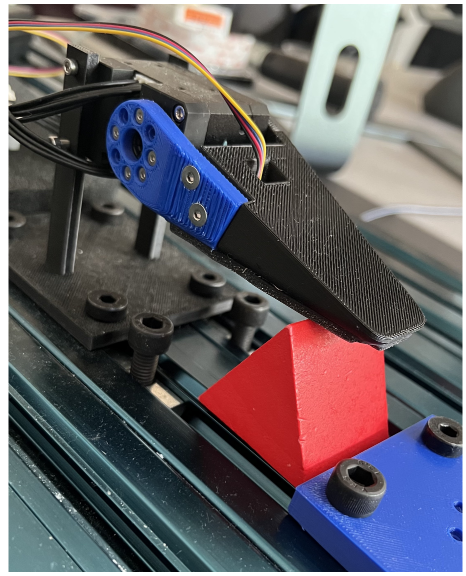


Fig. 2. finger policy being executed

elastomer with magnetic micro-particles and a ciruit board that contains several magnotrometers. When contact with an object is made, the elastomer conforms around it and changes the magnetic-field of the microparticles. The X,Y, and Z components of magnetic field changes are detected by the magnotrometers and reported by the ReSkin sensor.

IV. Experiments

Building up to the larger goal of using tactile sensing for dexterous manipulation, we began with some preliminary experiments to see how the ReSkin sensor performs at providing data for the classification of various shapes and textures.

A. Shape Classification

As a first experiment, a ReSkin sensor was attached to a 3D printed robot finger that was actuated by a dynamixel motor. See figure 1. A policy was written for the finger to apply a fixed torque for 5 seconds to an object and record ReSkin measurements of the change magnetic flux in the X,Y,Z coordinate system. See fig. 2. Six different objects were used in the shape classification study. The shapes classified were triangular prism, rhombohedron, pentagrammic prism, 4-leaf clover prism, and Hexagonal prism. See fig 3 Three labelled datasets were collected consisting of 30 taken trajectories for each of the 6 objects (25 training set sample and 5 validation set samples), i.e. 180 trajectories total for each dataset.

In the first dataset, in between each of the 30 samples, the object was translated to a different position so that contact could be made at different locations on the sensor. During the collection of the second dataset, the objects were both translated to a different position and horizontally rotated. In the third and final dataset, objects were translated, horizontally rotated as well as vertically rotated.

After data was collected, a 4-layer multi-layered perceptron

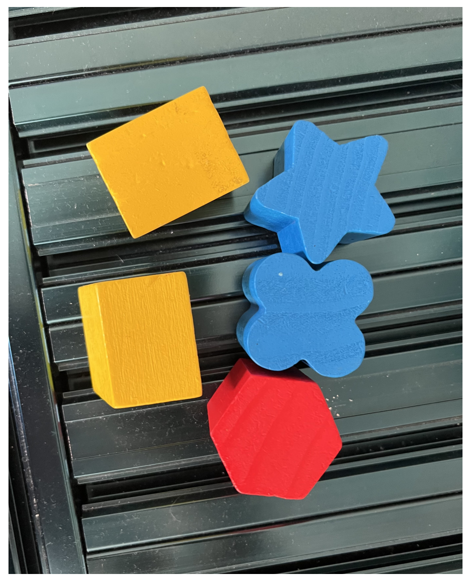


Fig. 3. Object shapes that were classified

model was trained on the labelled data to created a classification model. As can be seen in IV-A the accuracy of the classification model decreased as additional variance in the position and orientation of the objects were added. This demonstrates the need for large area sensing of objects. Since the finger only had limited contact area with the objects, as the number of different object orientations were introduced, the number of samples requires to maintain a certain level of accuracy increased. Therefore, for the same dataset size, we saw diminished classification accuracy.

Shape Dataset	Classification Accuracy
Translation	0.94
Translation + horizontal rotation	0.84
Translation + horizontal + vertical rotation	0.39

Fig. 4. Robot hand

V. Texture Classification

The next experiment conducted was a texture classification. In order to access the ReSkin sensor's ability to differentiate between object textures, an experiment was conducted where 6 identical balls were wrapped in various textures. The balls that were classified were: the original ball with no additional texture (Plain Ball), a ball wrapped in small bubble wrap (Small Bubble Ball), a ball wrapped in large bubble wrap (Big Bubble Ball), a ball covered in silicone sponge (Sponge Ball), a ball covered in cardboard (Cardboard Ball), and a composite ball that contained a combination of all the other materials used (Mixed Ball) See fig 4.

ReSkin sensors were attached to the fingers and palm of a three fingered robotic hand. See fig 6. A policy was written such that the ReSkin sensors would make contact with objects and "feel" various points of contact. See fig 5 A labelled data set was collected where the hand policy was run on each of the 6 different objects and ReSkin data was acquired. Then, similar to the first experiment, a multi-layered perceptron neural network was trained to classify the object's texture from the ReSkin data.

Texture	Classification Accuracy
Sponge Ball	0.93
Plain Ball	0.87
Big Bubble Ball	0.78
Cardboard Ball	0.76
Small Bubble Ball	0.59
Mixed Ball	0.38

VI. Conclusions and Future Work

This small study demonstrated that at least for small sets of objects, ReSkin sensor data can be used to differentiate objects based on their shape as well as their texture. This type of study can be expanded in many directions. Firstly, a larger study with more objects should be conducted to verify the results presented here. Secondly, studies can be performed to assess the ReSkin sensors ability to provide signal about other object properties: such as softness, density, Ferromagnetism, etc. In addition, work should be done to access how tactile data can be used in combination with already existing vision based systems to improve robot performance at various tasks.

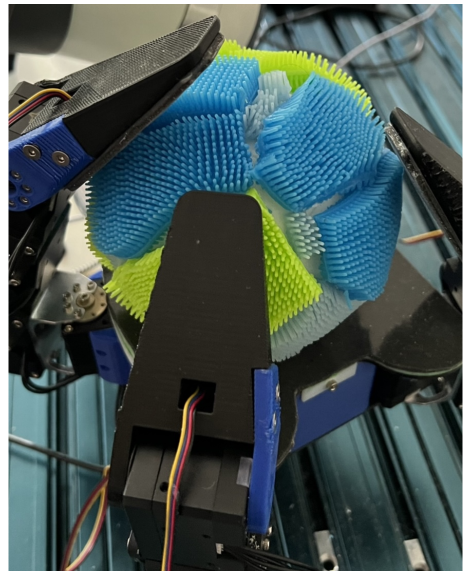


Fig. 5. Robot hand policy

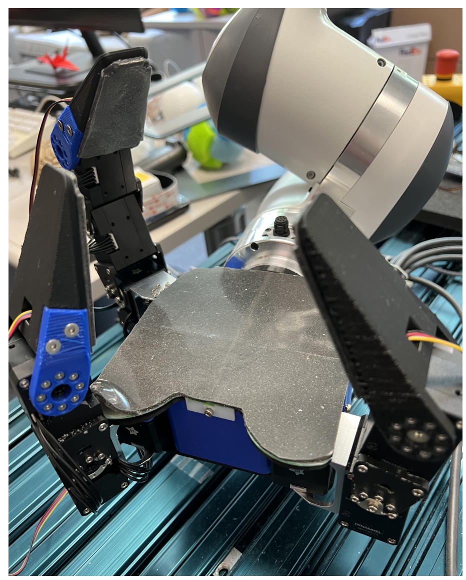


Fig. 6. Robot hand

VII. Acknowledgment

This work was supported by the School of Computer Science Robotics Institute. Thank you to Rachel Burcin and Dr. John M. Dolan their coordination of RISS and for making this research experience possible. A special thank you to Raunaq Bhirangi and everyone else at AGI Labs for guiding me through the summer.

REFERENCES

[1] R. Bhirangi, T. Hellebrekers, C. Majidi, and A. Gupta, "ReSkin: versatile, replaceable, lasting tactile skins," p. 11.

[2] OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning Dexterous In-Hand Manipulation," Jan. 2019, arXiv:1808.00177 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1808.00177

[3] O. Kroemer, S. Niekum, and G. Konidaris, "A Review of Robot Learning for Manipulation: Challenges, Representations, and Algorithms," Nov. 2020, arXiv:1907.03146 [cs]. [Online]. Available: http://arxiv.org/abs/1907.03146

[4] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, p. 2762, 2017.

[5] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, *et al.*, "Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3838–3845, 2020.

[6] A. Padmanabha, F. Ebert, S. Tian, R. Calandra, C. Finn, and S. Levine, "OmniTact: A Multi-Directional High Resolution Touch Sensor," Mar. 2020, number: arXiv:2003.06965 arXiv:2003.06965 [cs]. [Online]. Available: http://arxiv.org/abs/2003.06965

[7] T. Usui, H. Ishizuka, T. Kawasetsu, K. Hosoda, S. Ikeda, and O. Oshiro, "Soft capacitive tactile sensor using displacement of air–water interface," *Sensors and Actuators A: Physical*, vol. 332, p. 113133, Dec. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924424721005987

[8] Y. Li, L. Zheng, X. Wang, and W. Huang, "2 - Resistive tactile sensors," in *Functional Tactile Sensors*, ser. Woodhead Publishing Series in Electronic and Optical Materials, Y. Zhou and H.-H. Chou, Eds. Woodhead Publishing, Jan. 2021, pp. 13–30. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128206331000097

[9] Y. Zhang, F. Ju, X. Wei, D. Wang, and Y. Wang, "A Piezoelectric Tactile Sensor for Tissue Stiffness Detection with Arbitrary Contact Angle," *Sensors (Basel, Switzerland)*, vol. 20, no. 22, p. 6607, Nov. 2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7698970/

Predicting Tree Deformation under External Force with Graph Neural Networks

Jan-Malte Giannikos, Moonyoung Lee, Oliver Kroemer

***Abstract*— In light of labor shortages in agriculture, robotics can be an effective solution to automating labor-intensive tasks. However, for robots to robustly interact with trees in a closed-loop manner, an accurate and efficient model of tree deformation is required. Prior work proposed modeling tree dynamics as a spring-damper system, but such an approach introduces approximation errors and quickly grows computationally expensive for complex trees with a large number of branches. We instead propose to use Convolutional Graph Neural Networks (GCN) to learn the complex tree dynamics and predict tree deformation under external load. Preliminary work on this method showed promising results, but was limited by small datasets and therefore could not make strong claims about the generalizability of the model. We are using residual graph convolution networks with a linear stem and classification layer and adjustable depth. The models were trained and tested in simulation using synthetic datasets generated using the Space Colonization Algorithm (SCA). The proposed approach resulted in an average node position error of 0.01m when working with familiar tree topologies and an average node position error of 0.039m when working with unknown trees.**

As our testing shows, the proposed method of prediction can be applied to varying tree topologies, and we hope it will lead to safer and more accurate agricultural robots in the future.

***Index Terms*— Field Robots, Agricultural Automation, Data Sets for Robot Learning, Dynamics**

I. Introduction

The application of robotics in an agricultural context has greatly improved productivity and has reduced the need for menial human labor. However, some tasks could so far not be automated because they require interaction with plants, which are often complex in both shape and dynamics. Trees are a prominent example of this issue. More specifically, predicting how they will deform under external force is a difficult problem, that needs to be solved in order to create closed-loop systems that automate tasks such as pruning trees or harvesting fruit.

Crop manipulation tasks are usually modeled as a 3 stage process: Creating a 3D representation of the crop, modeling their physical properties, and finally planning a policy for interacting with the crops. Our work is targeting the second stage of the process, taking a graph representation of a tree and modeling its dynamic properties.

Our work proposes learning tree behavior with a Graph Neural Network (GNN). We model trees as a collection of tree points representing 3D joints that are connected through edges representing branch sections. We then translate this model into a graph representation where each node stores the physical properties of the tree. We also specify the external force acting on each branch segment as part of our graph. The graph representation is then fed into our GNN in order to predict the position of the tree under influence of the force.

In conclusion, our work provides:

- A novel approach to modeling tree deformation accurately by using a GNN
- A new synthetic dataset that simulates the physical deformation of trees under influence of external force.

II. Related Work

A. Crop Manipulation

Using 3D reconstructions of crops to reason about their behavior under external influence is a popular approach in Field Robotics and has been proposed by several papers [1] [2]. Yandun et al. [1] in particular used a spring damping model for joint connected branch segments to predict tree deformation under external force, an approach that inspired our physical simulation setup with which we generated our data-set. The Idea of modeling tree dynamics through joints with spring-damping coefficients was further reinforced by the results of Spatz and Theckes research into tree dynamics under the influence of wind [3]

B. Tree Generation

To generate a large data-set of tree deformations we need to acquire a large number of 3D tree models. However, creating such models from real tree data was deemed too costly. Therefore we decided to synthetically generate tree topologies with the Space Colonization Algorithm (SCA) presented in [4]. Branch radii were assigned using the pipe model from [5] and used to estimate joint stiffness.

C. Graph Neural Networks

GNNs are Neural Networks that operate on graphs by passing messages between their nodes. They were inspired by Sperduti et al. [6] who first introduced the idea of using neural networks on acyclic graphs. GNNs were first formalized by Gori et al. [7] and were further extended by [8] and [9].

Using GNNs to model physical Interactions between multiple entities is a technique that has been successfully applied by both [10] and [11].

In this project, we are using Graph Convolutional Neural Networks, since their application to modeling the deformation of simple tree topologies showed promising results in [12].

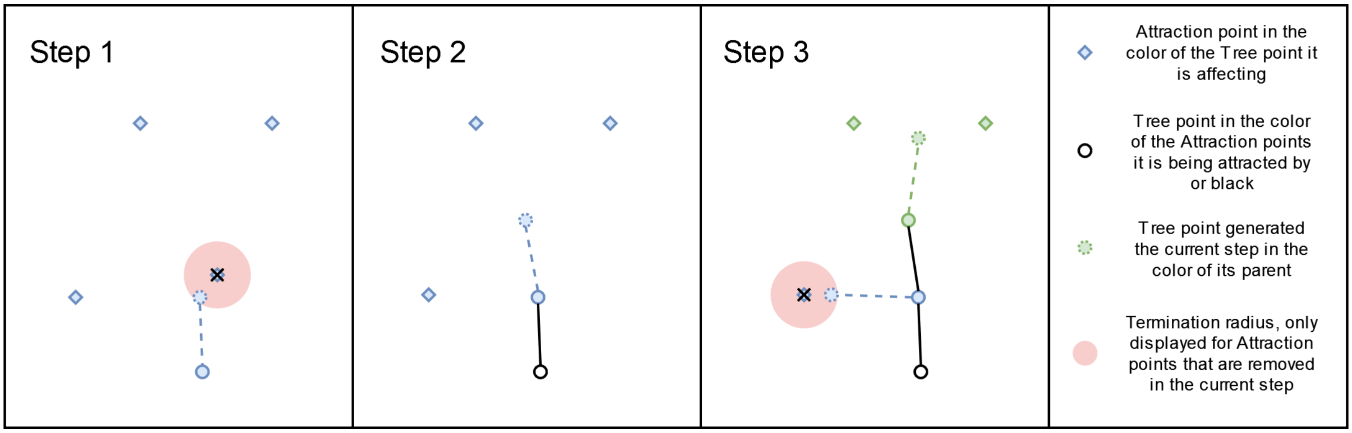


Fig. 1. Three example steps of the Space Colonization Algorithm

III. Dataset Description

To train a model, which generalizes well we need a sufficiently large dataset containing multiple different topologies. Since attaining such a dataset from real-world interactions is infeasible we instead opted for simulated tree-force interactions.

Each data point in our dataset consists of several positions, representing points in the tree (so-called tree points) that are connected by branch segments. We save these positions as well as the orientation of the connecting branch segments before and after the push occurred. We also save the external force acting upon each tree point.

The following sub-sections will explain the process of generating our synthetic dataset in more detail.

A. Tree Generation

We used the Space Colonization Algorithm (SCA) to generate tree topologies that were then used to generate a dataset of 10,000 pushes per topology in simulation. The SCA uses randomly distributed attraction points to simulate the competition for space between branches that occurs when trees are growing. Our implementation of the algorithm iteratively generates tree points until a user-defined number of tree points is reached. In each iteration it generates a child for each tree node that is being affected by at least one attraction point, as can be seen in Fig. 1.

Each attraction point affects only the tree point that is closest to it, while a tree point may be attracted by multiple attraction points. This means that we generate a set of relevant attraction points $S(v)$ for each tree point v. To calculate the position of a child, we use the following formula:

$$v' = \frac{\vec{n}}{\|\vec{n}\|} \cdot D \tag{1}$$

where

$$\vec{n} = \sum_{s \in S(v)} \frac{s - v}{\|s - v\|} \tag{2}$$

And where s is the position of the parent, v is the position of an attraction point. D is the distance between parent and child, given as a parameter of the SCA.

If a child comes within termination range of any attraction points those points are removed and can no longer attract any tree points.

After each round of tree point generation, we decreased the step size D. This allowed us to use long branch sections for the relatively homogenous tree trunk while simultaneously modeling the more complicated structure of the tree crown. This approach was crucial since we had a strong restriction on the number of tree points that could be imported into the simulation environment Isaac Gym [13].

We also introduced a new early stopping criterion, that allowed us to set the number of generated tree points exactly by potentially aborting the algorithm in the middle of a round of point generation as opposed to checking if the maximum number of tree points was reached only after each round was completed. Pseudo code for our version of the SCA can be found in the Appendix 1

After the general topology of the tree was created we calculated the radius of each branch segment by following the rule proposed in [5]. In short, branch segments without children were assigned a fixed tip radius, while all other branch segments were assigned a branch radius r based on the radius of their children $r_c \in R_c$ according to the following formula:

$$r = \sqrt[3]{\sum_{r_c \in R_c} r_c^3} \tag{3}$$

The stiffness s of each joint was determined based on the radius of the parent branch segment i.e., the branch segment that was closer to the root of the tree. The equation used to calculate the stiffness was the following:

$$s = C \cdot r^4 \tag{4}$$

Where C was a constant that allowed us to fine-tune the tree's behavior and the proportionality of s and r^4 was derived from the formula for the Modulus of Elasticity found in [14] under the assumption that a round branch segment could be roughly approximated by a square beam with $b = h$

B. Physical Simulation

To generate data for tree deformation under external force we used the Physx-based simulator Isaac Gym [13]. We imported the branch segments generated by the SCA as rigid bodies connected through 3D joints.

In cases where branching occurred all outgoing branch segments had separate joints, allowing them to move independently from one another.

We also added leaf segments that were used to track the tip position of every branch, since that information could otherwise not have been obtained directly through Isaac Gym. It should be noted, that the leaf segments were excluded when randomly choosing a segment to apply force on.

We then applied a random force (random both in magnitude and direction) to a random branch segment in the tree. Each force was constrained to a maximum magnitude of 17.3 Newton to prevent situations where the deformation of our tree would no longer be elastic in the real world. The force was applied until the tree stabilized i.e. stopped moving. We will refer to each of these force interactions as a ”push”. To generate positional data before and after each push, we saved the root position and orientation of every branch segment before and after the push had occurred.

After each push, we waited until the tree had reset to its original position and repeated the procedure.

We generated two datasets with the method described above. The first set contains 250,000 pushes on 25 tree topologies with 8 tree points each. The second contains 240,000 pushes on 24 tree topologies with 10 tree points each. Most of our work focuses on the first dataset with the second serving as a way to check how well our model generalizes to more complex tree structures.

IV. Methodology

Since we model our trees as branch segments connected by joints we can easily formalize our tree structure as a graph, and use the message passing performed by our Graph Neural Network (GNN) to model the interaction between two neighboring branch segments. We can therefore guide the interactions occurring between nodes of the network to follow physically realistic constraints (e.g. preventing non neighboring branch segments from directly interacting with one another). This also means that the method of transformation between our tree structure and the graph used by the GNN as well as the properties of each node in that graph, have a large impact on the performance of the model overall.

A. Graph Representation

Choosing a method to transform a given tree structure into a graph representation for our GNN was a non-trivial task, as several approaches were promising. Ultimately two graph formalizations emerged as viable, each with different advantages.

The first method aimed at predicting the final positions of each tree node directly, by representing each branch segment as a node in the graph. The parameters for each graph node of the input graph were the root position, the orientation and the force acting on the corresponding branch or leaf segment. These values were then used to predict the root position of each segment after the push had occurred.

The second method, instead attempted to predict the rotation of each branch segment to then reconstruct the tree from the root node up, calculating the position of each tree node in a post-processing step. In this graph formalization each branch segment was represented by a node with the following input parameters: root position, orientation, branch segment length, and force vector acting on it. Orientation was formalized as local rotation relative to the parent (the neighboring branch segment closest to the root of the tree) in quaternion. The rotation parameter was therefore representing what rotation would have to occur so that a hypothetical branch segment pointing in the direction of its parent would have the same orientation as the actual branch segment of the tree model. We then trained the model to predict the local orientation of each branch segment after the push.

This method allowed us to ignore most branch movements that were not directly caused by force since all orientations were relative to the branch segment's parent. Therefore orientation changes due to an orientation change of the parent would not have to be predicted by the network, but could instead be computed in post-processing.

As suggested in [12] introduced prior knowledge about the dynamics of the tree by making some graph edges unidirectional. All edges that were not part of the path between the force node and the root node were made unidirectional, pointing towards the branch tips, away from the bidirectional section of the graph. This structure allowed us to encode the assumption that branches that are not under the influence of force will not impact the deformation of the part of the tree that is. For the first graph formalization, we also chose the edge from the root node to its child to be unidirectional to encode the assumption that the position of the root node should never be changed, not even through force. Preventing message-passing into the root node means that no change in the state of any graph node can have an impact on the state of the root node, allowing us to enforce this assumption.

A side-by-side comparison for both graph formalizations for a given example tree can be seen in Fig. 2, where each node is visualized with its relevant features next to it. Note that there is an additional feature, which encodes the external force acting on a branch segment, that was left out of this figure for space reasons, instead, the node under force was marked red.

B. Residual Graph Convolutional Networks

Since the application of Residual Graph Convolutional Networks (RGCN) was shown to be successful on the

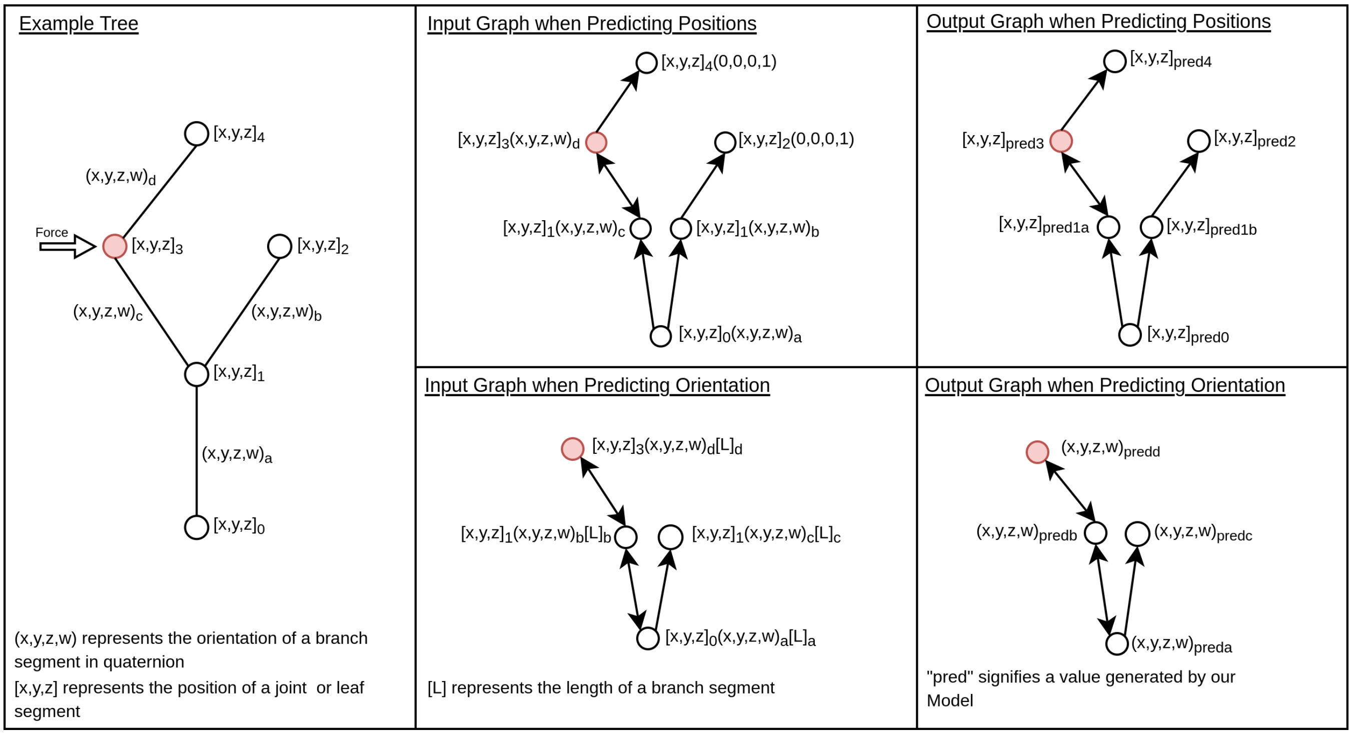


Fig. 2. The two graph structures for predicting orientation and position for an example push

problem in [12] we used the same model and architecture for this project. A Graph Convolutional Network (GCN) is a form of Message Passing Neural Network (MPNN) that was first introduced in [15]. It takes an input graph with parameters attached to each node and returns an output graph with the same structure. During inference time the network passes messages between each node and its neighbors in the next layer of the network. This means that the total number of message passing steps is dictated by the number of layers in the network.

A GCN takes an input matrix of node features X and an adjacency matrix A, representing the edges in the graph. X has the dimensions $N \times D$ where N is the number of nodes and D is the number of features per node, while A has the dimensions $N \times N$.

A GCN layer generally has the following structure:

$$H^{(l+1)} = f(H^{(l)}, A) \tag{5}$$

Where $H^{(0)} = X$ and $H^{(L)=Z}$ with Z as our output matrix and L as the number of layers in our network. We define the propagation function f as:

$$f(H^{(l)}, A) = \sigma(\hat{A}H^{(l)}W^{(l)}) \tag{6}$$

Where $\hat{A}$ is the normalized adjacency matrix with added edges from nodes to themselves, to allow for information retention in the nodes:

$$\hat{A} = D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}} \tag{7}$$

Where D is the diagonal node degree matrix and I is the identity matrix, used to add connections from each node to itself.

A Residual Graph Convolutional Network (RGCN) is a Graph Convolutional Network with residual connections. Each RGCN layer is constructed by concatenating a GCN layer with a linear layer and a skip connection.

C. Model Implementation

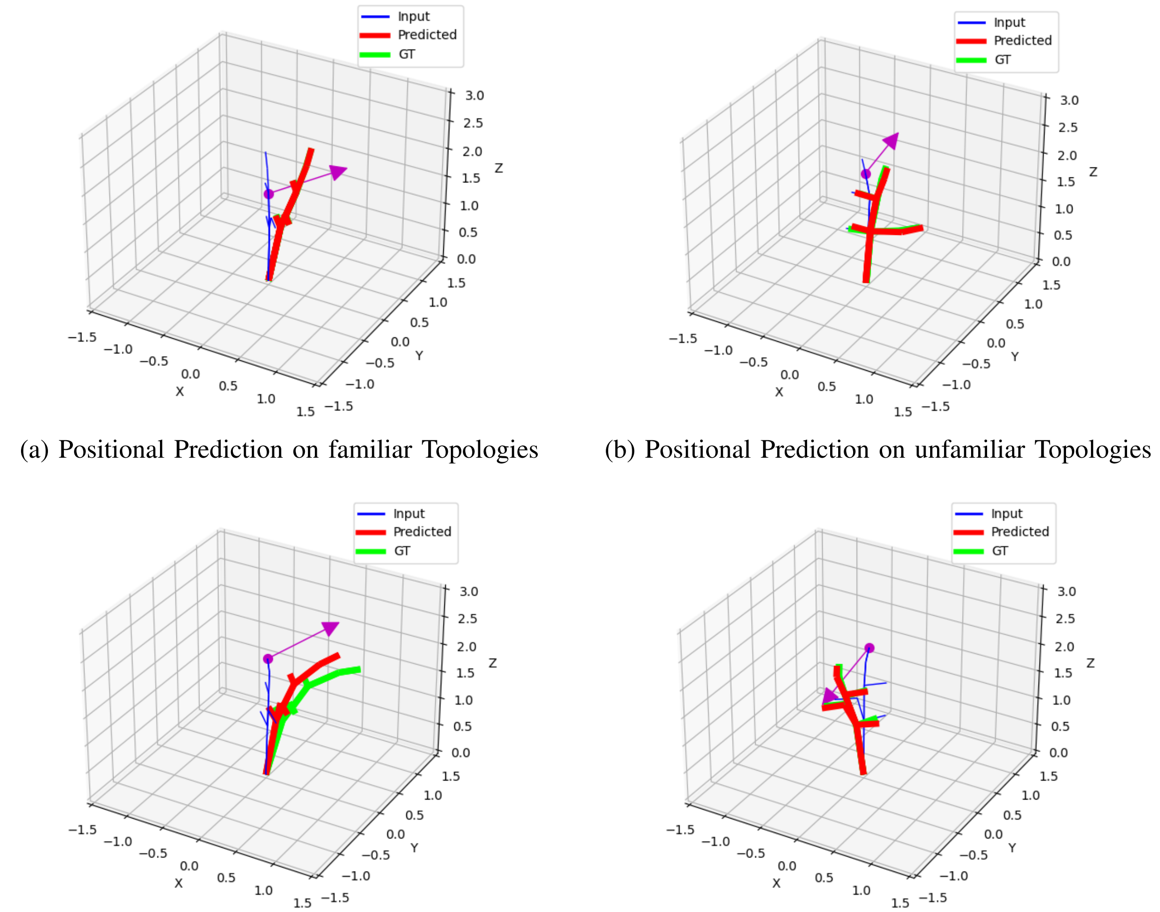
We used the GCN implementations from the libraries torch and torch.geometric as a baseline to implement our RGCN network.

We used a linear input layer to map our input data onto the RGCN layers, which each had 1280 node features and applied one-dimensional batch normalization [16] as well as dropout [17] after every layer, to prevent over-fitting. The number of layers was chosen, depending on the maximum tree size in the training dataset, by setting it to the number of nodes in the input graph. This allowed us to guarantee that message passing could occur between every node in the tree, even in worst-case conditions. The output layer was again chosen to be linear, mapping the output of the last RGCN layer to the output features determined by the respective graph formalization.

D. Loss Functions and Baselines

During training, we used the Mean Squared Error (MSE) as the loss on our first graph formalism, since we only had to optimize for distances between points. For the second formalism, we had to define a custom loss, since MSE was not viable for quaternion outputs. We chose the following loss function:

$$1 - (q_{pred} \times q_{gt})^2 \tag{8}$$



(a) Positional Prediction on familiar Topologies

(b) Positional Prediction on unfamiliar Topologies

(c) Orientational Prediction on familiar Topologies

(d) Orientational Prediction on unfamiliar Topologies

Fig. 3. Example Predictions for 8 Node Trees

Where q_{pred} is the normalized output quaternion and q_{gt} was the ground truth quaternion.

When evaluating we used the average node error e calculated with the help of the L2 Norm as our metric. It represents the Euclidean Distance between the predicted position of each tree node and the true position of the corresponding tree node averaged over every node n and every push p in the test data-set.

$$e = \frac{\sum_{p \in D} \sum_{n \in p} \|v_{pred} - v_{gt}\|}{|D||p|} \tag{9}$$

Where $\|\cdot\|$ is the L2 Norm, v_{pred} is the predicted position of a given node n and v_{gt} is the true position of a given node n.

In order to make grounded claims about our models performance we also established a baseline b to compare it to. Since no other sophisticated models had been trained on this novel dataset we chose to simply compare ourselves to the average total displacement. This baseline was calculated for the entire test-set D containing 2000 pushes p:

$$b = \frac{\sum_{p \in D} \sum_{n \in p} \|v_{before} - v_{after}\|}{|D||p|} \tag{10}$$

Where n is a single graph node, $\|\cdot\|$ is the L2 Norm, v_{before} represents the relevant output parameters of n before the push and v_{after} represents them after the push.

In order to be able to fairly compare two models with each other, even if the average total displacement of their test set is different we also provide e's percentage of overall displacement r:

$$r = \frac{e}{b} \cdot 100 \tag{11}$$

V. Results

When evaluating the two graph structures on our data-set it quickly became clear that the predictions were much more accurate if the model had been trained on the tree topology it was evaluated on.

A. Training and Evaluating on Identical Topologies

Predicting positions directly as well as predicting orientations before reconstructing positions achieved satisfactory results, as can be seen in I. It is clear that predicting positions outperforms predicting orientations by a significant margin, making it the preferred choice, when the tree topologies the model will be deployed on are known.

TABLE I

PERFORMANCE OF GRAPH FORMALIZATIONS ON FAMILIAR AND UNFAMILIAR TOPOLOGIES

Model	Average Node Error	Baseline Displacement	Percentage of Overall Displacement
Positional prediction on familiar topologies	0.011m	0.126m	8.6%
Orientational prediction on familiar topologies	0.042m	0.145m	29%
Positional prediction on unfamiliar topologies	0.039m	0.123m	31.7%
Orientational prediction on unfamiliar topologies	0.074m	0.14m	52.9%
Positional prediction on trees with higher complexity	0.082m	0.134m	61.3%
Orientational prediction on trees with higher complexity	0.114m	0.145m	79.1%

It should be noted that when predicting orientation the node error increased towards the tips of the tree, most likely due to error propagation: Since the post-processing step reconstructs the tree from the root up, errors in the prediction for nodes close to the root propagate into nodes further out. This effect can be seen in 3(c)

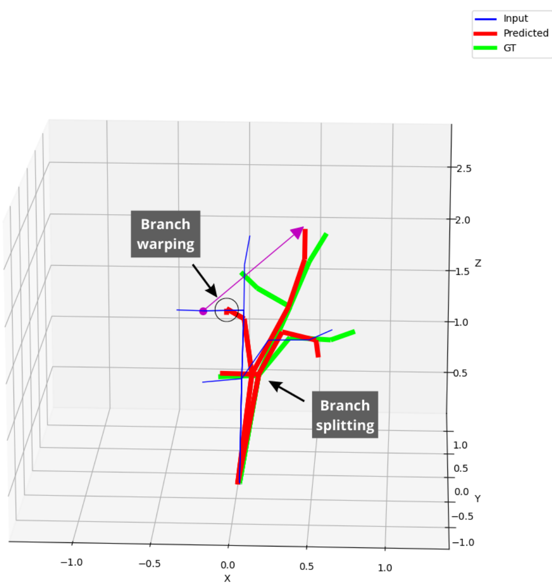


Fig. 4. Example of the two forms of tree warping when using positional prediction

B. Training and Evaluating on Seperate Topologies

Since it is unrealistic to assume the tree topologies our model will be deployed on are known during training time unless the field robots perform some form of online learning, we also tested our models on tree topologies that did not occur within the training set.

When evaluating on topologies that were not trained on we can see a sharp drop in performance, as evident in Table I. Predicting positions still performs better, but when looking at plotted predictions a weakness of the positional approach becomes apparent. Since all positions were predicted independently of one another, the model is able to change the distance between parent and child nodes, shortening or lengthening branch segments. Additionally since some node positions are duplicated (as can be seen in 2) the model can theoretically predict different positional values for these nodes leading to branch splitting.

Both of these effects lead to tree warping that fundamentally changes the tree topology, an issue that cannot come up when predicting orientations.

C. Evaluating on More Complex Topologies

To get an idea of how well our model generalizes to more realistic trees with a higher number of tree points we also tested models trained on trees with 8 tree points on trees with 10 tree points. As expected performance decreased again, however, it did not become arbitrarily bad, showing that a limited level of generalization is possible with the applied model architecture.

The issues of tree warping when using positional prediction described in the previous subsection became even more pronounced. An example can be seen in 4.

VI. DISCUSSION

While predicting positional data seems to be the overall most accurate method, its tendency to warp the tree topology when interacting with unknown and more complex tree structures was deemed to be a critical flaw. Predicting the orientation might therefore be preferable, especially in environments in which keeping the tree topology consistent is important, like for example the planning of longer policies. Additionally, errors stemming from error propagation along the tree structure are easier to reason about than seemingly random warping errors. Generally, our experiments showed, that the application of GCNs on this task can be successful, even when predicting tree deformations on topologies that had not been trained on. This leaves us hopeful, that generalization to trees with different levels of complexity is also possible. However, we also realized that there are inherent limitations to GCNs that we cannot overcome. Most prominently the commitment to a fixed number of layers at training time makes it difficult to generalize to trees with a significant jump in complexity compared to our training data. Using more flexible graph-based learning algorithms, therefore, seems to be necessary to successfully solve the larger problem at hand.

APPENDIX

Input: max_tree_pts, $termination\_radius$;
$attraction\_points \leftarrow$ random attraction points;
$tree\_points \leftarrow$ [initial tree point] ;
$edges \leftarrow []$;
while *number of attraction points* > 0 **do**
    for $v \in tree\_points$ **do**
        $S(v) \leftarrow$ attraction points that v is closest tree point to;
        $child \leftarrow$ generate child based on $S(v)$;
        $tree\_points$ **append** $child$;
        $edges$ **append** $(index(v), index(child))$;
        for $a \in attraction\_points$ **do**
            if $\|a - child\| < termination\_radius$ **then**
                remove a from $attraction\_points$;
            end
        end
        if $len(tree\_points) >= max\_tree\_pts$ **then**
            stop;
        end
    end
end

Algorithm 1: Space Colonization Algorithm

ACKNOWLEDGMENTS

I would like to thank Rachel Burcin and John M. Dolan for organizing the Robotics Institute Summer Scholars Program, which made this work possible. I also want to thank Moonyoung Lee and Oliver Kroemer from the Carnegie Mellon's Intelligent Autonomous Manipulation (IAM) lab for guiding and advising me throughout the project. A special thanks to the DAAD for funding this research and all members of the IAM lab for being welcoming and supportive.

REFERENCES

[1] F. Yandun, A. Silwal, and G. Kantor, "Visual 3d reconstruction and dynamic simulation of fruit trees for robotic manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[2] S. Katyara, F. Ficuciello, D. G. Caldwell, F. Chen, and B. Siciliano, "Reproducible pruning system on dynamic natural plants for field agricultural robots," in *International Workshop on Human-Friendly Robotics*. Springer, 2020, pp. 1–15.

[3] H.-C. Spatz and B. Theckes, "Oscillation damping in trees," *Plant science*, vol. 207, pp. 66–71, 2013.

[4] A. Runions, B. Lane, and P. Prusinkiewicz, "Modeling trees with a space colonization algorithm." *NPH*, vol. 7, no. 63-70, p. 6, 2007.

[5] K. Shinozaki, K. Yoda, K. Hozumi, and T. Kira, "A quantitative analysis of plant form-the pipe model theory: I. basic analyses," *Japanese Journal of ecology*, vol. 14, no. 3, pp. 97–105, 1964.

[6] A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 714–735, 1997.

[7] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings. 2005 IEEE international joint conference on neural networks*, vol. 2, no. 2005, 2005, pp. 729–734.

[8] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[9] F. M. Bianchi, C. Gallicchio, and A. Micheli, "Pyramidal graph echo state networks," in *ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2018.

[10] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia, "Learning to Simulate Complex Physics with Graph Networks," in *Proceedings of the 37th International Conference on Machine Learning*. PMLR, Nov. 2020, pp. 8459–8468, iSSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v119/sanchez-gonzalez20a.html

[11] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende, *et al.*, "Interaction networks for learning about objects, relations and physics," *Advances in neural information processing systems*, vol. 29, 2016.

[12] H.-J. Huang, C. H. Kim, and M. Lee, "tree is all you need," 7 2022. [Online]. Available: https://github.com/chjohnkim/tree_is_all_you_need

[13] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[14] H. Zhang, J. F. Hunt, and L. Zhou, "Comparison of wood composite properties using cantilever-beam bending," *BioResources*, vol. 10, no. 2, pp. 3070–3078, 2015.

[15] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[16] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

Treating Video as an Image

Jana Zeller[1], Deva Ramanan[2] and Jonathon Luiten[3]

***Abstract*— Many applications for dense video understanding task require harsh memory or time constraints, like Autonomous Vehicles or mobile Augmented Reality. Past research tried to address the problem by either down-sampling the resolution of inputted video or by trying to slim / prune single pass models. While these solution lead to reasonable results, most of them do not take advantage of the similarity of consecutive frames in video. In order to address this issue we are working on Treating Video as an Image. In this paper we show, that our proposed method is significantly faster than just running a vision transformer on video and that our method has the potential to significantly decrease compute time with only minor drops in accuracy.**

I. INTRODUCTION

Dense video understanding tasks, such as Video Object Segmentation, Video Instance Segmentation [1] and Video Panoptic Segmentation [2] have recently found use in a number of applications, for example in Autonomous Vehicles (AV) [3] and Augmented Reality (AR). Oftentimes these applications require strict time and/or memory constraints, e.g. AVs demand real-time segmentation while mobile AR relies on memory-efficient solutions. Even if there are no strict constraints most common offline approaches manage only to analyse videos of between 6 and 10 seconds at a frame-rate of around 6 fps [1], [4], since anything beyond this dramatically increases the compute time and the required GPU memory. This significantly slows down research trying to learn long-term behaviour from videos or using Computer Vision for Video Editing or Augmented Reality.

Most state-of-the-art approaches for video understanding rely on architectures that have been constructed for images, that were then adapted to video by incorporating temporal context [5]–[7]. Generally these approaches process the majority of the video on a frame-by-frame basis. For instance, Mask2Former for video understanding [4], [7] runs the backbone and the pixel decoder once per frame. Only the transformer decoder is run once per video, but according to our measurements the transformer decoder only accounts for 21% of the whole compute time. Note that this measurement has been taken with a relatively simple R50 backbone and that this number would shrink further if we would use a more complex R100 or Swin-Transformer backbone (see fig. 1).

We hypothesize that this frame-by-frame inference introduces a lot of unnecessary overhead, mostly since consecutive frames in a video are highly correlated. Therefore we introduce **"Treating Video as an Image" (TVI)**.

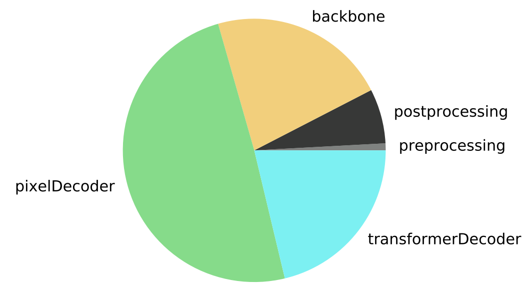


Fig. 1. Compute time of Mask2Former with a R50 backbone. The backbone and the pixel decoder are computed on a frame-by-frame basis.

Inspired by DataMUXing in Neural Networks [8] TVI processes multiple frames simultaneously by the same backbone and pixel decoder. We achieve this by simply concatenating consecutive frames together as though they were a single frame with more input channels. The key to allowing this architecture to be useful for dense prediction tasks is a *'video expansion' layer* which, after processing the combined frames, extracts dense features for each frame individually. This way *TVI* learns which features should be shared between frames and which features should be extracted per frame.

For simplicity we demonstrate in this paper how TVI enhances the Mask2Former architecture with a R50 backbone.

II. RELATED WORK

A. Speedup in a given model

There are two main approaches to gain speed-up in video understanding. Either sub-sampling the input video, in a spatial or temporal way, or optimizing the model itself.

Sub-sampling the input's frames has been a common approach to optimize *global* video analysing tasks, like action recognition. Salient clips are significant snippets of a video, that when analysed *increase* the accuracy of action recognisers compared to analysing the whole video [9]. "Dynamic images" take the idea of sub-sampling even further. They condense a whole video into a single frame and then only classify this image [10]. While these approaches show significant speed-up for global task, they are not applicable for dense task, because dense tasks require labels

[1]Jana Zeller is with Karlsruhe Institute of Technology jana.zeller@student.kit.edu This work was written during an internship at Carnegie Mellon University.

[2]Deva Ramanan is with Carnegie Mellon University and ARGO AI deva@andrew.cmu.edu

[3]Jonathon Luiten is with RWTH Aachen and Carnegie Mellon University. jluiten@andrew.cmu.edu

for each and every pixel. Thus, only looking at every x-th frame significantly decreases performance, whereas action recognition can already be solved reasonably well when only looking at a single frame [11].

However, a question arises: Is it possible to apply a similar approach to dense tasks? Since simply skipping on frames does not yield reasonable performance (see fig. 6), a possible answer has to be incorporated within the model's design.

Optimizing the model: There have been a multitude of endeavours trying to create better speed/accuracy/memory trade-offs for object detectors, however most of the current research focuses on *single-pass detectors* [12]. A common way to create speed-up in single-pass detectors is to prune or slim models. This approach is backed by the *lottery ticket hypothesis*. This hypothesis states that only a fraction of each neural network is relevant to solving a given task [13]. However, there is debate regarding whether pruning models actually improves performance compared to randomly initializing weights [14].

Multi-Input-Multi-Output Networks: While leaning down models is one way to interpret the lottery ticket hypothesis, the hypothesis also backs *Multi-Input-Multi-Output* networks. The main idea behind these networks is that multiple inputs can be mixed and passed through a neural network once. Afterwards the output can be "unmixed" again, generating per instance labels. This approach has been applied to both CNNs [8], [15], [16] as well as Vision Transformers [17]. However, most prior research focuses on mixing *unrelated* images together [8], [17] or on mixing the same image in order to achieve a better accuracy [15], [16].

With TVI we hope to show, that mixing related frames from the same video can outperform current State-of-the-Art approaches on accuracy-latency-trade-offs.

B. Current Vision Transformers

While TVI has the potential to speed-up *any* offline Vision Transformer for video that runs on a frame-by-frame basis, in this working paper we only demonstrate how TVI can speed up Mask2Former with a R50 backbone.

Mask2Former for Video Instance Segmentation is a Vision Transformer built for images, that has been adapted to work on video.

The original Mask2Former architecture for images consists of a backbone, pixel decoder and a transformer decoder. First the *backbone* extracts low-resolution features. The *pixel decoder* then takes these features as input and gradually scales those up to achieve *per-pixel high resolution embeddings*. A *transformer decoder* in return converts those embeddings to *object queries*. The final binary mask is a result of combining the per-pixel embeddings and the object queries. For more detail please refer to [4].

To adapt this architecture to video the transformer decoder is run once per video instead of once per image. This way the transformer decoder can attend to 3D spatial-temporal features instead of attending to 2D spatial features. This way the transformer decoder can track objects across space and time. For further information please refer to [7].

Because the adapted architecture is so similar to the architecture used for image segmentation tasks, it can highly benefit from pre-trained weights for images. This and the small youtube-vis dataset (2k videos) results in little needed training time (8k iterations for the youtubevis 2021 dataset with a batch size of 16). During training each video is composed of 2 randomly sampled frame in a 20 frame range. Different randomly created transformation are applied for each of those two frames. This leads us to believe, that there is a opportunity to gain even more performance by incorporating more context and taking more advantage of the highly correlated data.

TVI has the potential to fill this gap. Because TVI is trained in a teacher-student fashion, TVI can be trained on unlabeled data. This allows TVI to be trained for longer and to actually learn which features should be shared between consecutive frames. This way TVI incorporates context, while being faster than other state-of-the-art architectures.

III. METHOD

We make two major adaptions to the Vision Transformer architecture used by Mask2Former: 1) We merge images in the stem of the backbone and 2) we expand the outputted features of the pixel decoder to multiple frames (see fig. 2).

We treat videos as a volume of dimension $T \times C \times H \times W$, where H and W denote the width and height of each individual frame and T the time, i.e. the total number of frames in the video and C the feature dimension, e.g. 3 for RGB colored videos. Let M denote the number of frames which we want to merge together.

A. MERGING FRAMES

During pre-processing we concatenate M frames together using the C dimension. Thus we convert our video of dimensions $T \times C \times H \times W$ to dimensions $T/M \times C*M \times H \times W$.

We also adjust the *stem*, i.e. the first convolution in the backbone, to expect $C*M$ input channels instead of C. This way, the 2D convolution can attend to all 3 spatial-temporal dimensions, while only being as compute-expensive, as a 2D convolution. We initialize the weights by simply concatenating the original weights together using the feature channel C and then dividing by the number of merged frames M.

B. EXPANDING TO VIDEO

The original pixel decoder returns multiple volumes of dimensions $C \times \hat{H} \times \hat{W}$, where C denotes the number of feature channels after both the backbone and pixel decoder have been applied and $\hat{H}$ and $\hat{W}$ denote the corresponding post-processing height and width.

We take each of these volumes and pass them through a simple linear, one-layer neural network which expands $C \times H \times W$ to $M \times C \times H \times W$. We initialize the weights using a simple copying function.

IV. EXPERIMENTS

We evaluate TVI on the Mask2Former architecture on the Youtube-VIS 2019 dataset. We used a R50 backbone.

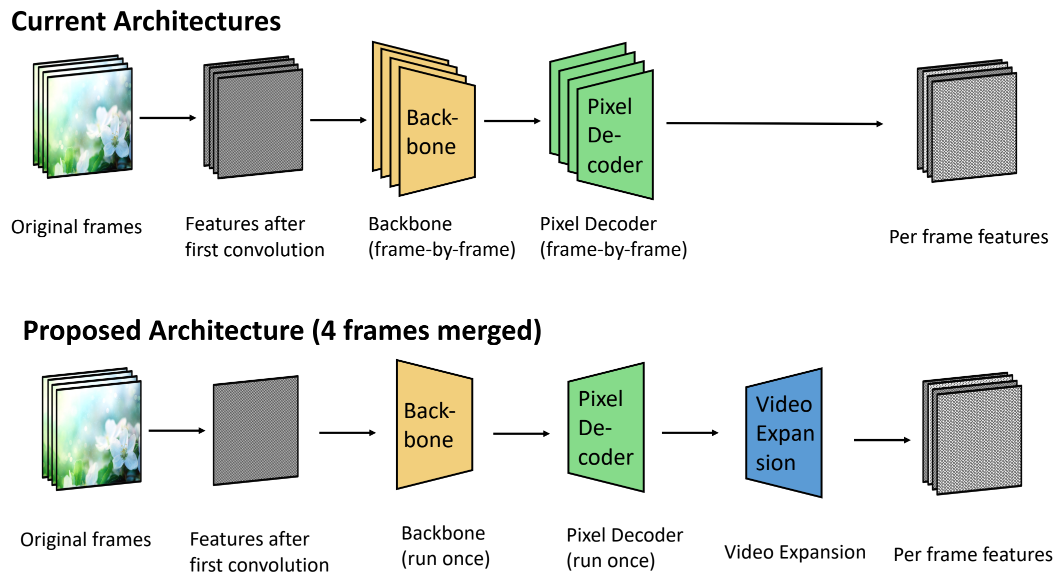


Fig. 2. Proposed Architecture

A. SPEED UP

All in all, we only altered the first convolution and added very simple, fast networks in order to get per frame features. Thus, we only need to process the backbone and pixel decoder roughly *1/M*-th of the time compared to Mask2Former (see fig. 3). The added overhead from the expansion and merging step are only minor in comparison (see fig. 4). All in all, we achieve a major speed-up the more frames we merge (see fig. 5).

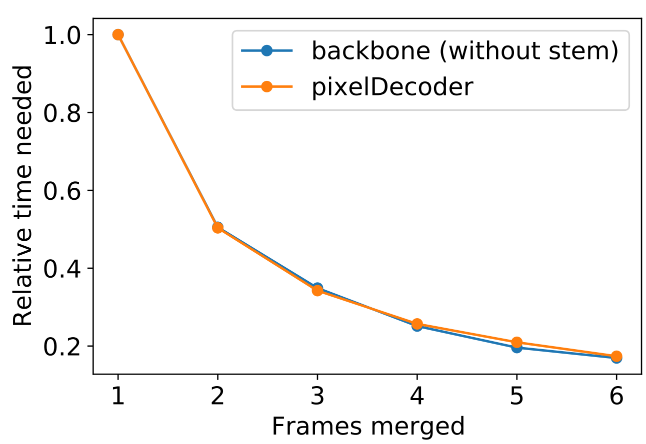


Fig. 3. Relative total compute time for 200 videos (5,000 frames) during inference

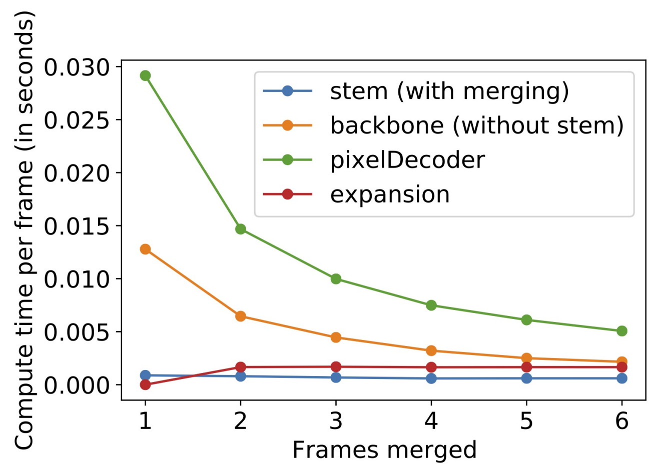


Fig. 4. Absolute total compute time per frame per stage during inference

B. PERFORMANCE

We trained our model by sampling at least M, i.e. the number of merged frames, frames in a consecutive fashion from each video. We applied the same transformation proposed in Mask2Former for all frames in a video, while making sure that the identical transformation are applied to each frame in the same video. All of this allows our custom backbone and pixel decoder to learn temporal features from consecutive frames.

We first trained on the mask loss proposed by Mask2Former on the youtube-vis 2019 dataset. This however yielded poor performance. We hypothesize that this is because the youtube-vis 2019 dataset only consist of 2k training videos, which are not enough to learn temporal features.

V. FUTURE WORK

We plan to train our model in a student-teacher-learning manner. We want to train using videos from the WebVid Dataset [18], which we preselect on the basis of whether their video description contained one of the object categories

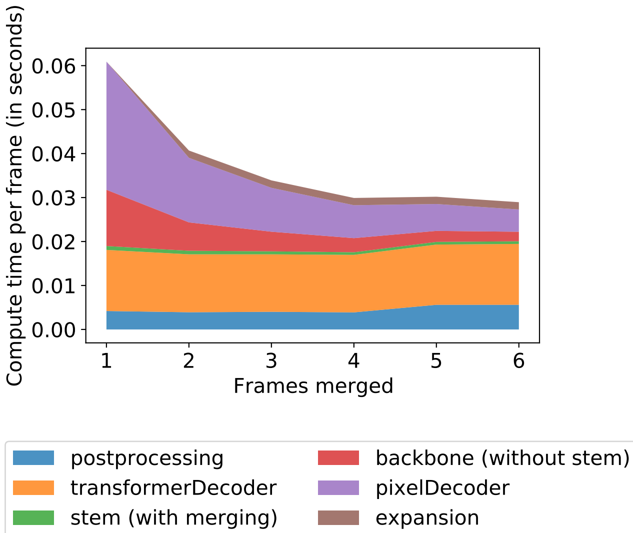


Fig. 5. Relative total compute time for 200 videos (5,000 frames) during inference

of the youtube-vis 2019 dataset. We want to first run the original model for each frame of the video and use then the original high-resolution per-pixel embeddings outputted from the pixel-decoder as our target.

Finally we want to compare our method against an interpolation-based pixel-wise nearest neighbour baseline (see fig. 6).

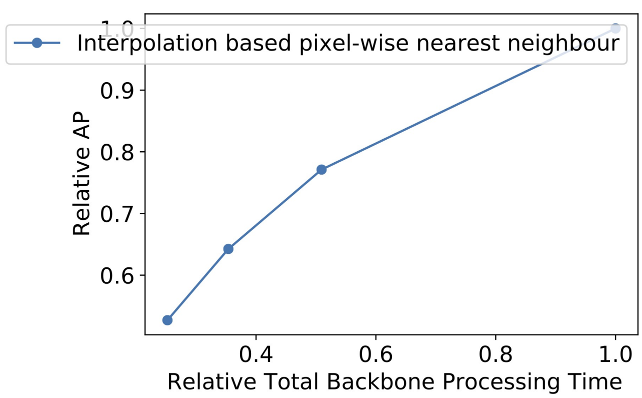


Fig. 6. A baseline based on frame-subsampling for dense video understanding tasks

We want to explore whether we can improve performance further. Either by fitting our model after it has been trained on WebVid to the Youtube-vis dataset or by first computing the loss through the output of the pixel-decoder and then moving to compute the loss using the outputted masks.

We currently hypothesize that the performance of TVI is more limited by the time between the first and last frame, rather than the number of frames merged. In order to test our hypothesis, we want to compare how TVI performs on 6fps videos and 30 fps videos at different merge rates.

Moreover, we want to show that TVI cannot only speed-up Mask2Former with a R50 backbone, but also different transformer based backbones, like Swin and other offline architectures. We also want to have a look at how much less memory TVI uses compared to the original architecture.

Finally, we also want to explore if and how much memory TVI uses compared to the original Mask2Former architecture.

VI. Conclusion

We present a new way to process video for dense video understanding tasks. Treating Video as an Image has the potential to create massive speedups with only minor drops in accuracy.

ACKNOWLEDGMENT

This work was supported by Argo AI Center for Autonomous Vehicle Research. Jana was supported by the German Academic Exchange Service. The authors would like to thank Rachel Burcin and John M. Dolan for their support. They organized the Robotics Institute Summer Scholar Programme during which this work was written.

References

[1] L. Yang, Y. Fan, and N. Xu, "Video instance segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[2] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, "Video panoptic segmentation," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9856–9865, 2020.

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[4] B. Cheng, A. Choudhuri, I. Misra, A. Kirillov, R. Girdhar, and A. G. Schwing, "Mask2former for video instance segmentation," *arXiv preprint arXiv:2112.10764*, 2021.

[5] S. Hwang, M. Heo, S. W. Oh, and S. J. Kim, "Video instance segmentation using inter-frame communication transformers," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 13 352–13 363.

[6] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia, "End-to-end video instance segmentation with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 8741–8750.

[7] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," *CoRR*, vol. abs/2112.01527, 2021. [Online]. Available: https://arxiv.org/abs/2112.01527

[8] V. Murahari, C. E. Jimenez, R. Yang, and K. Narasimhan, "Datamux: Data multiplexing for neural networks," *arXiv preprint arXiv:2202.09318*, 2022.

[9] B. Korbar, D. Tran, and L. Torresani, "Scsampler: Sampling salient clips from video for efficient action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[10] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould, "Dynamic image networks for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[11] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[12] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," *CoRR*, vol. abs/1611.10012, 2016. [Online]. Available: http://arxiv.org/abs/1611.10012

[13] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Training pruned neural networks," *CoRR*, vol. abs/1803.03635, 2018. [Online]. Available: http://arxiv.org/abs/1803.03635

[14] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, "Rethinking the value of network pruning," *arXiv preprint arXiv:1810.05270*, 2018.

[15] S. E. Ahmed Raza, L. Cheung, D. Epstein, S. Pelengaris, M. Khan, and N. M. Rajpoot, "Mimo-net: A multi-input multi-output convolutional neural network for cell segmentation in fluorescence microscopy images," in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, 2017, pp. 337–340.

[16] S. E. A. Raza, L. Cheung, D. Epstein, S. Pelengaris, M. Khan, and N. M. Rajpoot, "Mimonet: Gland segmentation using multi-input-multi-output convolutional neural network," in *Medical Image Understanding and Analysis*, M. Valdés Hernández and V. González-Castro, Eds. Cham: Springer International Publishing, 2017, pp. 698–706.

[17] R. Sun, C. Masson, N. Thome, and M. Cord, "Adapting multi-input multi-output schemes to vision transformers."

[18] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, "Frozen in time: A joint video and image encoder for end-to-end retrieval," 2021. [Online]. Available: https://arxiv.org/abs/2104.00650

Object Manipulation with Simple Box Gripper

Jessica Ahner[1] and Oliver Kroemer[2]

***Abstract*— The creation of advanced robotics systems with the ability to manipulate small objects, bringing them to precise positions and orientations, has proved to be an important, yet challenging, task in the world of robotics. Manipulation of small items is typically done with complex dexterous manipulators, using fingertips that require fragile strategies given the constraints of having to maintain contact. In this paper, we propose a box gripper design for robust in-hand manipulation without the complexity of common approaches. We run a series of experiments testing the picking up, manipulation, and placement of a variety of objects. We do this by utilizing extrinsic dexterity with the box gripper implemented on the Franka Panda Robot Arm. The goal of this work is to develop a general understanding and background on the manipulation capabilities of a simple box gripper and how the capabilities vary with objects of different size and shape.**

I. INTRODUCTION

In today's world of fast technology, it is important to consider the tradeoff between time spent creating a design and functionality. Although complex dexterous manipulators may be more successful at specific functions, we must ask if the time and money required to create these manipulators are actually necessary to complete the task. Many studies have shown that simple hands still possess many usable manipulation capabilities. Our goal in this paper is to find an effective way to complete a broad task without the same complexities as the dextrous hand.

Our approach is just the beginning of more well-rounded future projects. We will be using a controlled environment with known locations and orientations of the small objects that will be manipulated. It is important to start small, learning the foundation of the experiments, before adding more complexities like irregular shaped objects. With a very simple box gripper design, it is easy to make small edits and create new iterations in a short amount of time. This allowed for numerous variations to be created and either tested or disposed of based on whether or not they appeared viable for the experiments. These grippers have no sensors or actuators and rely completely on extrinsic dexterity to create movement and assortment.

When it comes to extrinsic dexterity, most previous experiments have used fingers to aid in the use of these extrinsic forces, however, we will be using an even simpler method for our object manipulation. With the use of the tilt caused by gravity, we will experiment the abilities of a simple box gripper to move, maintain control of, and orient different objects. These broad manipulation capabilities may not appear complex enough to be necessary, but these simple tasks are still extremely desired and useful.

Fig. 1. Box Gripper Design

The central goal of this paper is to examine and further prove the idea that simple manipulators like a box gripper can perform important tasks without the added complexity or time and money requirements of more complex manipulators. In our experiments, we run several trials of picking up, orienting, and relocating items through the use of our box gripper on a Franka Panda Robotic Arm. From this data, we can assess the success of simple manipulators, addressing the following objectives:

-Pick and Place
-In-hand Movement
-Multi-Object Grasping
-Simple Gripper Design

Section III walks through our approach for creating and implementing the box gripper design as well as our desired path of motion for the robot to complete each objective listed above. Section IV describes our experimental setting and the specific actions we took to evaluate the abilities of the box gripper. This is also where we discuss the results obtained from the different experimental trials based on each

[1]Jessica Ahner is with the Electrical/Computer Engineering Department, United States Air Force Academy, Colorado Springs, USA and Robotics Institute Summer Scholars, Carnegie Mellon University, Pittsburgh, USA. c23jessica.ahner@afacademy.af.edu

[2]Oliver Kroemer is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, USA. okroemer@andrew.cmu.edu

objective. Finally, section V discusses the impact of the results as well as the improvements that can be included as future work is conducted based around these ideas.

II. Related Work

In this section, we explore related work on previously researched object manipulation by simple robotic systems. This work involves the use of extrinsic dexterity, box manipulators, and basic anthropomorphic hands.

Many approaches to object manipulation involve complex robotic hands, using intrinsic dexterity to move and reorient an object. This strategy has proven to be expensive in time and materials. The related work provided shows the broad and quite beneficial abilities of using less complex designs in order to accomplish similar tasks of object manipulation.

Most similar to the experiments that will be completed in this paper is Erdmann and Mason's exploration of robotic manipulation through motion strategies without the use of sensors. [1] Through a planned path, objects can be oriented by tilt to achieve a desired position. These experiments required restrictive assumptions that, if not met, ultimately resulted in the object maneuvering in an unplanned path. This research presents how nontrivial sensorless manipulation can be. However, the manipulation does not involve collection and placement of these oriented objects. This idea will be explored with our implementation of the box gripper.

Other approaches involve in-hand manipulation through use of extrinsic dexterity. This means the objects are manipulated with forces such as tilt, gravity, squeezing, rolling, and many other techniques that are extrinsic to the hand. Nikhil Chavan Dafle and others studied a series of regrasp actions with sufficient successful experiments to show that in-hand manipulation is possible even on very simple hands. [2]

Another related research area revolves around the benefits and abilities of a simple hand design when compared to a more complex design. These simple hands have less actuators and sensors, but are therefore lighter and less expensive. Matt Mason and others explore the trade off between simplicity in design and robotic hand function. Although there were high success rates with bin-picking tasks, there was noise due to high clutter, pose uncertainty, and overlapping fingers. These are the concerns with simple hands, however machine learning and manipulation of error and confidence parameters helped Mason allow for more accurate detection of object singulation. [3] The box gripper explored in our experiments will further work to represent how simple manipulators can accurately detect position and pose with an even simpler design than the simple hand.

III. Approach

The central aspect to our approach is the design and measurements of the box gripper. In order for the experiments to be effective in testing the success of the pickup, organization, and placement of objects, the box gripper design must be accurate. Once a gripper design is created, we can explore the best sequence of movement on the Franka Robot to test the success parameters.

A. Gripper Design

First, we perfected and created the box gripper design, including the box and the scooping mechanism. Based on the dimensions of the Franka Robot and the desired dimensions of the box, two different designs were presented and ultimately adjusted numerous times. The initial box design shown did not allow the scooper to easily slide under the box and it did not provide sufficient stability when attached to the robot. The second design improved in both these areas and ultimately was the final design. With two supports to connect to the robot, a compartment under the box for the scooper to easily slide, and an elongated scooper to provide more scooping ability, the second design shown would allow for better experimentation.

Fig. 2. Box Design Variations

B. Action Path

Our approach to adequately assessing the success of the box gripper would be through picking up multiple objects, storing them in the back compartment of the box, bringing all objects back to the front of the box, then releasing them in a precise location. This would allow us to evaluate a variety of different motions in relation to the different objects we were testing. Now that we had decided what our decided actions for the gripper were for our experiments, we needed to design an adequate sequence of motions on the Franka Panda Robotic Arm. To approach this, we began by testing for a small yet significant enough tilt angle to sort the objects without losing the inability to pick up more objects. The tilt angles for many of the motions ended up different because some in-hand manipulation inside the box required a harsher tilt than others. We also needed to analyze the precise picking up and dropping off height and location for the robot arm in order to ensure successful trials. These things were necessary to examine before implementing the box gripper to prevent inaccurate and potentially destructive motion of the box gripper on the robotic arm.

With the desired path, pickup and drop off heights, and tilt angles figured out, we could code the process for the robotic arm to complete. We created numerous functions for the Robot including tilting, closing hands, opening hands, and we also included a neutral position for the arm to return to after each task. We realized this was necessary to avoid disruption in the joints and to get the robot to reach precise locations. It was expected that the desired path would not completely match the actual path implemented, so the code was designed with the intent of being easily adjusted and modified.

The primary goal of our approach to this design is to create a solid foundation to run the following experiments. It was important to have a stable and durable box design with an adequate path of motion to allow for grasping, in-hand movement, and relocation.

IV. Experiments

A. Setup

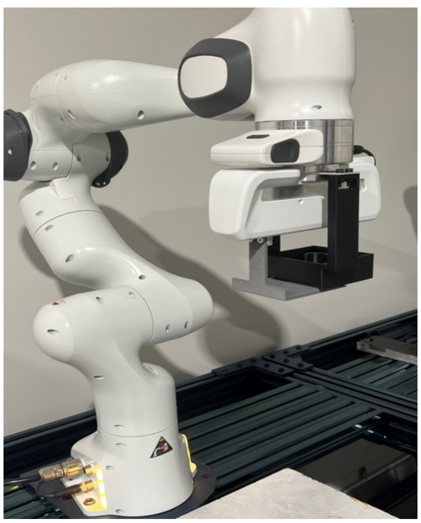


Fig. 3. Box Gripper on Franka Panda Robot

In the experimental set up, the box gripper was attached to the Franka Panda Robot Arm as shown in figure 3. The objects will be picked up and placed on the same area of the platform for each trial so that we can maintain a controlled environment. The platform sits at a slight angle which is accounted for in the code. We have chosen numerous different objects to test against the preprogrammed plan that evaluates the objective of this experiment. The variety of objects help show what characteristics of the objects affect the success of the gripper. We are testing rolling objects, larger objects, rectangular shaped objects, objects that wont slide as well in the box, and objects that do not have symmetrical designs like screws. These objects are shown in Figure 4.

Fig. 4. Trial Objects

Through these experiments we ran the same program on the Franka Robot Arm. This program would next perform a series of tilts that were established in the approach section in order to move the objects picked up in the front of the box to the back of the box behind the wall inside the box. Once this was completed, the robot would one by one, pick up more objects, each time placing the object behind the wall before picking up the next one. We ran the program 10 times for each object, ultimately testing the success and consistency of the program with the different sizes and shapes of objects.

B. Pickup Results

A rectangle was marked on the platform to indicate the size of the opening of the box gripper. As long as the objects were placed inside the marked area on the table, the box gripper would be able to pick them up. Any items placed outside that area will not be successfully picked up since they would not be located inside the box when the box approached the table. The only issue with this was that the platform was not flat, so there was a slight gap between the box and the table on one side, causing certain objects to squeeze through and not get picked up. This had to be accounted for in the code so we created a slight tilt in the box as it approached the table to pick up items. With the slight slope on the edge of the scooper, the round objects tested were easily slid on top of the scooper and into the box. However, the rigid objects failed to be picked up most of the time. Completely rigid objects with no rolling capabilities failed each trial and could not be picked up. Objects with slight rolling capabilities, but rigid characteristics failed when placed in certain orientations, but were successful when picked up at other orientations. Objects with complete rolling capabilities, like the marble, were successfully picked up on each of the 10 trials.

C. Manipulation Results

The first manipulation experiment conducted was the movement of objects from the front of the box to the back for storage. The storage of items in the back of the box would allow for pickup of more items in the front without spilling the previously picked up objects out of the box. In Figure 5, the sequence of tilts can be seen that move the objects from where they are picked up in the front to the back. Similar to the results of the pickup operations, completely rigid objects could not adequately slide and therefore could not successfully be moved around in the box. The Lego block failed each time in moving to the back of the box. Objects that we partially round and partially rigid, had successes and failures moving throughout the box. These objects included the die and the screws. Fully round objects like the marbles successfully completed the motion on every trial.

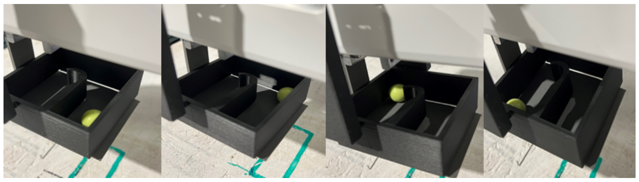


Fig. 5. First Sequence of Motion

The second portion of the experiment consisted of picking up multiple objects, storing each one in the back of the box before picking up the next object. Then, the objects were repositioned to the front of the box, together, in order to precisely place them back on the platform. The reasoning for the curved edge in the middle of the box was to keep the objects in the back from rolling forward when more objects were being picked up. Figure 6 shows the sequence of movements to move all the objects to the front of the box. The results for this portion were the same as the results for the first manipulation sequence as they both had to do with the movement of objects through the box.

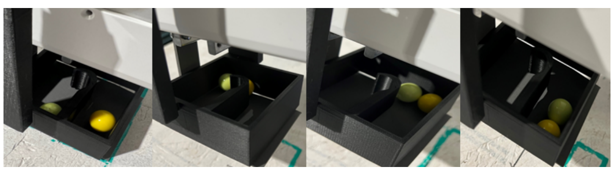


Fig. 6. Second Sequence of Motion

D. Placement Results

The placement trials demonstrated a lot of variance with the results. The placement trials with the die were successful and easy to observe, but the trials with round objects were much more difficult to observe due to the slanted platform. Figure 7 shows a few examples of precise placement withe the 10-sided die. No matter where the die were to begin with, the die ended up in the same place after the manipulation in the box. However, when these trials were done with round objects, they were placed in the correct location, but they would roll down the platform because of the slope. This made precise result collection difficult for the more rounded objects. In each of the trials with the round objects like the marbles, we found that no matter where the marbles were picked up from, they would be placed in the same precise location when they were released.

V. Conclusion

Our experimental trials indicate the usefulness of a simple box gripper to complete broad tasks such as picking up, manipulating, and precisely placing objects down on a surface. Through use of extrinsic dexterity, we were able to find a level of functionality without the expensive and complex nature of dexterous hands and other more complex designs.

This project exists at a very elementary stage, as many desired changes to allow for better experimentation were not possible. The experiments were used to build a foundational understanding of a simple box gripper and to learn the limitations that would come with the simple design. There was small room for change during the experimentation process, therefore if we came across small errors, we could not easily go back and change the entire process. We found the results for each object were relatively consistent across the three objectives. Rigid objects were incredibly difficult to manipulate due to the added friction. The more round characteristics that an object possessed, the easier it was to manipulate inside the box. These limitations are things that should be explored in future work in order to find ways to overcome them.

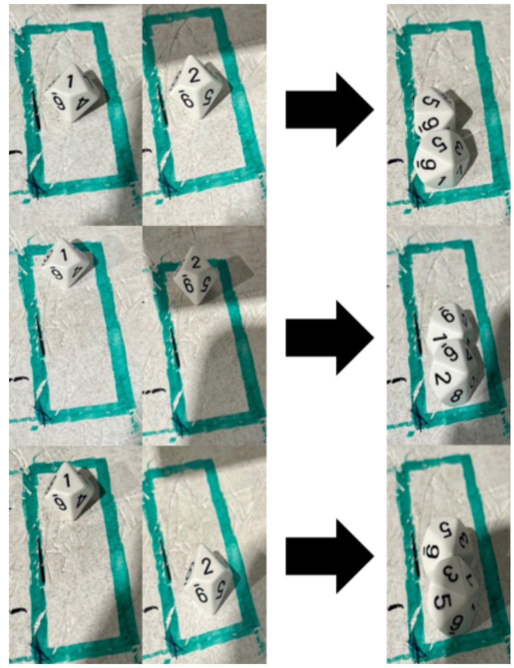


Fig. 7. Precise Placement of Die

A. Future Work

This project requires more time and resources than were provided in this initial exploration. Future work should target one specific area of simple object manipulation and deconstruct it in order to further explore these ideas with more detail. First, I will explain my recommendations for improvements as this project is continued. Then, I will elaborate on different projects that may stem from this elementary research.

The most important things to improve would be the experimental setup and design quality. The platform utilized in this experiment was slanted, therefore altering our results. With a flat surface, there would have been less roll off from the objects and outcomes of the placement would be easier to examine. For design quality, I would recommend creating a box gripper out of a material with less friction and ridges. Creating the box gripper on a 3D printer caused difficulty for many of the trial objects to slide around inside the box. A flat metal would be a great alternative.

Further experimentation can involve box gripper with task specific barriers for more accurate sorting of objects. Adding a second compartment in the back of the current box gripper would allow for more precise storage and manipulation of more than two objects at one time. Exploring new potential strategies of in-hand manipulation with the box gripper is another great path to pursue. For example, using the box gripper to orient a screw onto its flat head to limit rolling would increase manipulation capabilities. These further experiments should also involve the use of different materials to build the box gripper in order to evaluate how the capabilities change with each box. These can include metal, wood, and more cohesively built plastic materials.

Acknowledgments

This work would not have been possible without the support of the Robotics Institute Summer Scholars Program. A large thanks to Dr. Oliver Kroemer from the Carnegie Mellon's Intelligent Autonomous Manipulation lab for his amazing mentor-ship and support throughout the program. I would also like to thank Rachel Burcin and Dr.John Dolan for their coordination and assistance throughout the summer. Lastly, I am incredibly grateful for The United States Air Force Academy to allow me the opportunity to forgo a period of military training in order to pursue my passions in the field of academia.

References

[1] M. Erdmann and M. Mason, "An exploration of sensorless manipulation," *IEEE Journal on Robotics and Automation*, vol. 4, no. 4, pp. 369–379, 1988.

[2] N. C. Dafle, A. Rodriguez, R. Paolini, B. Tang, S. S. Srinivasa, M. Erdmann, M. T. Mason, I. Lundberg, H. Staab, and T. Fuhlbrigge, "Extrinsic dexterity: In-hand manipulation with external forces," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1578–1585.

[3] M. T. Mason, A. Rodriguez, S. S. Srinivasa, and A. S. Vazquez, "Autonomous manipulation with a general-purpose simple hand," *The International Journal of Robotics Research*, vol. 31, no. 5, p. 688–703, 2011.

[4] A. Rodriguez, M. T. Mason, and S. S. Srinivasa, "Manipulation capabilities with simple hands," *Experimental Robotics*, p. 285–299, 2014.

[5] A. Holladay, R. Paolini, and M. T. Mason, "A general framework for open-loop pivoting," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3675–3681.

[6] A. S. Morgan, K. Hang, W. G. Bircher, F. M. Alladkani, A. Gandhi, B. Calli, and A. M. Dollar, "Benchmarking cluttered robot pick-and-place manipulation with the box and blocks test," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 454–461, 2020.

Automated Content Editing in NeRFs

Joel Julin[1,2], Heng Yu[2], László Jeni[2]

Abstract— **Neural rendering for novel view synthesis has been a rising problem within the computer vision community. Among the many proposed techniques, neural radiance fields (NeRF) have proven to be one of the most effective. The system introduced by NeRF builds a 3D representation of a scene given a number of 2D images. When applied to dynamic scenes, NeRF's performance significantly declines. Recent strides have been made towards solving this problem with the likes of CoNeRF and Non-Rigid NeRF; both works have shown to be effective in re-rendering and manipulating neural radiance fields despite the presence of dynamic objects. However, this previous research is hindered in both the labor and function domain. CoNeRF requires the tedious task of manually annotating the dynamic component of the input images; whereas Non-Rigid NeRF is unable to generalize to new movements and only works with a single deformable object. We propose a followup method capable of re-rendering and manipulating a dynamic object within a radiance field without the need for manual annotation. With our proposed method, dynamic scenes with simple movement of the human shape (i.e. raising/lowering of the arm) can be more easily rendered and manipulated through automatic masking of the component in motion. In addition to dynamic scenes, our work also brings benefits to static scene manipulation through selective ray rendering that allows for entire removal of humans from the scene, or inversely, the removal of the background. We hope that this work sheds light on future NeRF manipulation methods.**

Index Terms— **Computer Vision, Neural Rendering, Computer Graphics**

I. INTRODUCTION

Neural Radiance Field (NeRF) has recently become the standard method for view synthesis. This is certainly not without reason, as NeRF has outstanding performance on both static [1]–[5] and dynamic objects [6]–[9]. Despite this performance in both domains, radiance field manipulation and usability remains an open research question. There is a large amount of research dedicated to the application of NeRF to dynamic scenes, but it is often tedious and requires much work.

One such work that aims to control the dynamic movement within neural radiance fields is CoNeRF. CoNeRF yields impressive results, but its scalability is largely limited due to the need for manual annotation of the controllable component. There also exist few methods that utilize automatic segmentation of the dynamic and static components but fail to provide fine-tuned rendering selection, such as Non-Rigid NeRF [6]. In this paper, we propose a method capable of automatically, and accurately, annotating both a dynamic and static human body for use in neural radiance field control and manipulation. For use in dynamic scenes, our method proves to be effective at radiance field control with CoNeRF. Within the static domain, the same method is effective at segmentation and rendering control. This work predominantly showcases efficient usage of image segmentation for dynamic scene control while also revealing an additional use-case for static scenes. In specific, this paper describes a method that offers:

- **Automatic Selection of the Human Shape for Neural Rendering.** Automatically selecting the shape of interest within a scene brings heaps of improvements to both static and dynamic neural rendering, namely the elimination of manual annotation. While currently structured to solely segment human bodies, this work can be extended to a variety of other classes.
- **Dynamic Scene Manipulation.** Many of the NeRF methods that apply to dynamic scenes [9] require a degree of manual annotation or are limited in function [6]. This work applies automatic segmentation methods to this dynamic scene control.
- **Static Scene Manipulation.** Given the selected object(s) the neural radiance field can be rendered without the selected objects, or without the background.

II. RELATED WORKS

Our work is closely related to a number of recent developments made within neural rendering.

A. Neural Rendering for Novel View Synthesis

NeRF has led to cascades of research on neural rendering for novel view synthesis. The original method proposed by Mildenhall et al. [1] is capable of building a high fidelity 3D representation of a scene given a number of 2D images. When initially published, a large constraint of this method was its inability to represent non-rigid or dynamic scenes. Since then, there have been a few works that extend NeRF's exemplary performance on static objects to objects in motion [6]–[9].

B. CoNeRF: Controllable Neural Radiance Fields

CoNeRF is one of the most influential NeRF followup works that propose a method for neural radiance field control [9]. The method proposed in this work controls object movement through tedious annotation of the controllable component (i.e. arm moving) and a value assignment. To annotate the controllable component, CoNeRF uses a manual click and drag annotation software known as labelme [10]. At each annotated frame, a value is assigned to represent

[1]Joel Julin is with the School of Computing and Information, University of Pittsburgh, Pittsburgh, PA 15260, USA jmj96@pitt.edu

[2]Joel Julin, Heng Yu, and László Jeni are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15260, USA jjulin, hengyu, laszlojeni@andrew.cmu.edu

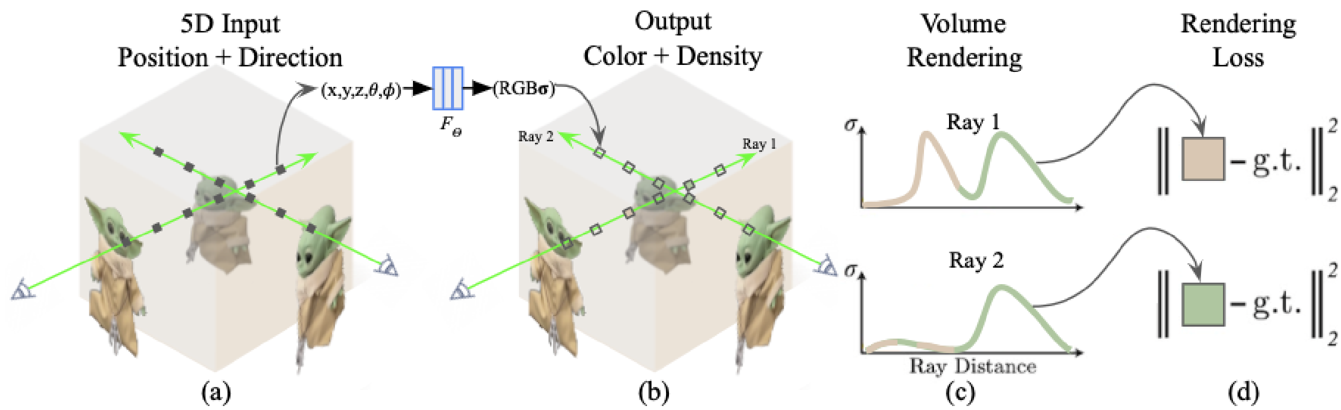


Fig. 1. NeRF architecture

the transition status. For instance, an arm fully located at a person's side would be assigned a value of -1.0, an arm that is fully stretched out would be given a value of 1.0, and an arm that is somewhere in the middle would be assigned a value of 0.0. Since our work brings automatic annotation to CoNeRF and improves its usage, it is closely related. However, our work does not overlap with that of CoNeRF's [9] in terms of contribution. We only use this preexisting work as a means to display our application of automatic annotation.

C. *Non-Rigid NeRF*

Non-Rigid NeRF [6] focuses on the automatic separation and manipulation of a scenes rigid (static) and non-rigid (dynamic) counterparts. Non-Rigid NeRF is largely limited in function, as this work *only* achieves scene manipulation when composed of both static and dynamic objects. Our work significantly extends upon theirs since we enable radiance field manipulation regardless of the scenes composition; meaning that distinct dynamic and static components are not required.

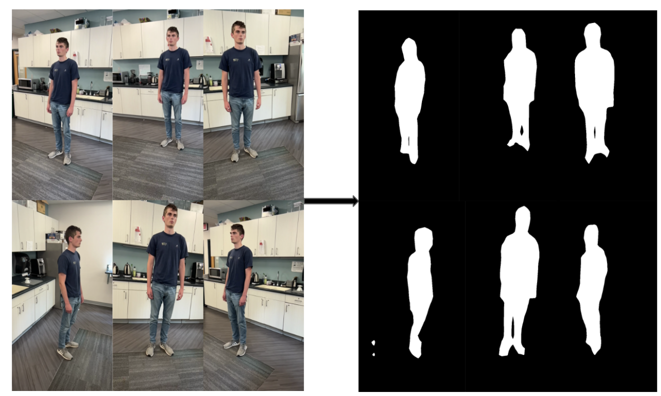
III. METHOD

Our method consists of four components (i) data collection and preparation, (ii) automatic segmentation, (iii) NeRF architecture for automated content editing, and (iv) automated segmentation for dynamic scenes.

A. *Data Collection and Preparation*

The data used for this work was captured using an iPhone 13 Pro's 240fps slo-mo camera. For a thorough 3D representation of the scene and a large number of viewing angles, the video was captured in a circular motion with a moving camera. After the video is captured, a sparse set of the captured frames (approximately 300) are passed through COLMAP to obtain the camera poses that are needed to determine where the camera is located during each frame. This is important because without the poses, there is no structural information of where these images were captured in relation to other images, and are needed as inputs to NeRF's fully connected network that will later be discussed in further detail.

B. *Automatic Segmentation*



a) input b) masked output

Fig. 2. Body Pix 2.0 generates near perfect masks. These binary masks are used to determine the object of interest. Areas marked as white are editable and areas marked as black are unaffected. It is important to note that these masks can be inverted. When this happens, the areas marked as white become black and the areas that were once marked as black become white, effectively switching the areas of interest.

BodyPix 2.0 is an effective segmentation software that is directly trained to recognize, and segment, the human shape. As shown in Fig. 2, the masks that BodyPix 2.0 creates are quite accurate. The masks generated by this software are used to determine which object in our scene we wish to manipulate.

C. *NeRF Architecture for Automated Content Editing*

At the heart of this work is the standard NeRF method. [1]. This method works by representing a static scene as a fully-connected deep network with a 5D coordinate (x, y, z, θ, ϕ) and an output of color and density (RGBσ). As shown in Fig. 1, given an image from a given viewing direction or camera position (θ, ϕ) a ray is passed through each pixel. As that ray is sent through the pixel at location (x,y), a sampled point z is sent through a fully connected deep network and is outputted a color (RGB) and density (σ) where density is a value that denotes whether or not an object is present.

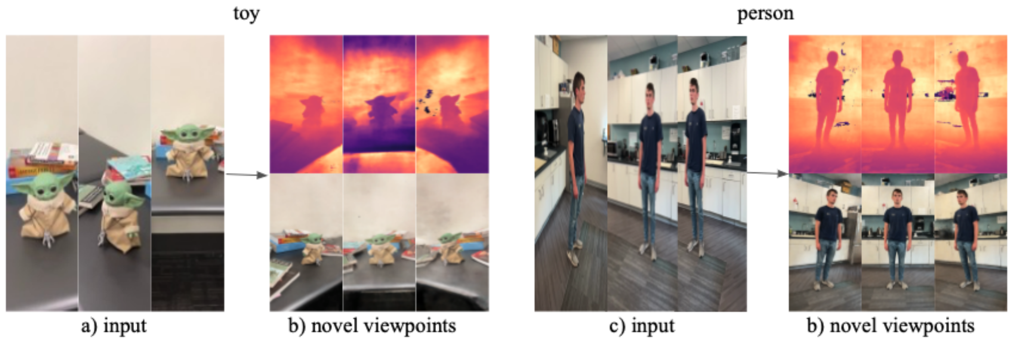


Fig. 3. The standard NeRF method yields impressive results when applied to our own image sequences. The inputs to both of the above experiments contain approximately 150 images from a variety of different viewing directions.

This process of sampling a point along the ray and passing it through the deep network to receive an output of color and density is repeated for every sample along the ray as shown in part b of the figure. Whenever this process is completed, all of the sampled points are combined using a classical volume rendering technique [11] to receive the final prediction of the pixel's color as shown in part c. The final step (d) within the NeRF architecture is to compute the loss between the rendered color and the ground truth, then take that loss to reduce the rendering error in future iterations.

With the standard NeRF method now being outlined, we will now present our simple, yet successful, modification that allows for automated content editing. Since NeRF renders the color of a pixel from a sampled ray, removing the entire ray effectively prevents portions of a scene from being rendered. When this ray removal is applied at a larger scale, by utilizing selective binary masking, entire objects can be removed from the scene. Partnering binary object masking with ray removal is the extent of our method that allows for static scene manipulation.

D. Automated Segmentation for Dynamic Scenes

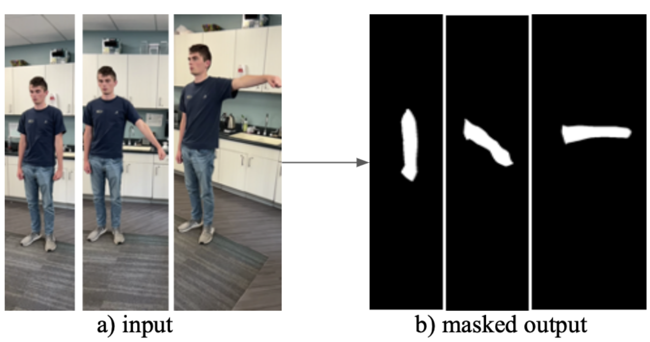


Fig. 4. Body Pix 2.0 is capable of automatically generating masks for parts of the body, such as an arm.

Our method of utilizing automatic segmentation for static NeRFs, also brings benefits to NeRFs representing dynamic scenes. Instead of removing components of a scene, we applied this method to more easily control a scene. CoNeRF [9] acted as our standard dynamic method to which we made modifications to. This method uses manual annotation to signify which component within a scene is in motion, which is oftentimes a tedious task. The automatic segmentation software, Body Pix 2.0, entirely removes the need for manual annotation of the controllable segment of the scene as shown in Fig. 4. The primary modification made to CoNeRF was to directly accept binary images as labels instead of a json file containing the mask coordinates generated by labelme [10] (a manual annotation software). Other than this modification, the original CoNeRF code was used for our experiments.

IV. EXPERIMENTS

The experiments that were conducted for this work include standard NeRF without modifications acting as our baseline, static NeRF modifications with both manual and automatic annotations, and dynamic NeRF modifications with automatic annotation. In the following sections we will explain each of these experiments in complete detail.

A. Baseline: Standard NeRF

This first experiments conducted for this project used the standard NeRF code without any modifications. The NeRF architecture, as shown in Fig. 1, sends rays through every pixel from a given viewing direction, samples along each of the rays to obtain (RGBσ), and applies a volumetric rendering technique [11] to accumulate the sampled points and render each pixel's predicted color. The loss between the ground-truth and this prediction is used to reduce the rendering error in future iterations. We applied this method to two of our own image sequences, a toy and a person (Fig. 3), with each sequence containing approximately 150 images.

B. Static NeRF Modifications

The two experiments for static NeRF modifications concerned object and background removal. For both of these experiments, the standard NeRF model was modified such that certain rays falling within a masked region are not rendered.

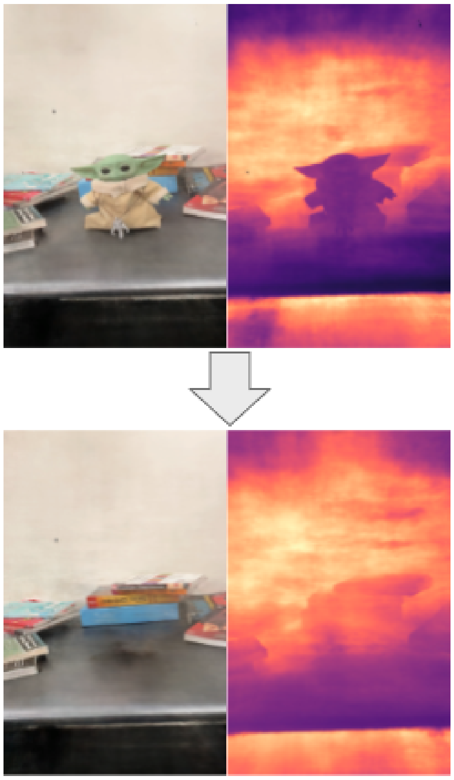


Fig. 5. Our method is capable of fully removing an object from a neural radiance field (bottom). This successful removal becomes even more apparent when compared to the standard NeRF output and depth map generated from the same images (top).

1) Object Removal: Our method for object removal was first tested on a manually annotated image sequence as shown in Fig. 5. As expected, manual annotating a large amount of images is tedious. Nevertheless, this experiment demonstrated the effectiveness of our method before introducing automatic annotations from Body Pix 2.0.

The success of this method continues to hold when using Body Pix 2.0's automatic annotation to generate binary masks. As shown in Fig. 6, the person is removed from the neural radiance field. By using a much more efficient annotation method, these experiments become much more feasible.

2) Background Removal: Similar to the object removal experiments, we first observed the result of our background removal method using a manually annotated scene before using an automatically annotated one. Fig. 7 showcases a viewpoint taken from this experiment. Since annotations for our project are binary (i.e. selected component is marked as white and unaffected as black), it took little modification to our object removal method to remove the background. Using the automatic annotations from Body Pix 2.0 we can more easily remove the background from NeRFs, as shown in Fig. 8.

C. Dynamic NeRF Modifications using CoNeRF

In order to apply our usage of automatic segmentation to CoNeRF, a NeRF variant that aims to control dynamic

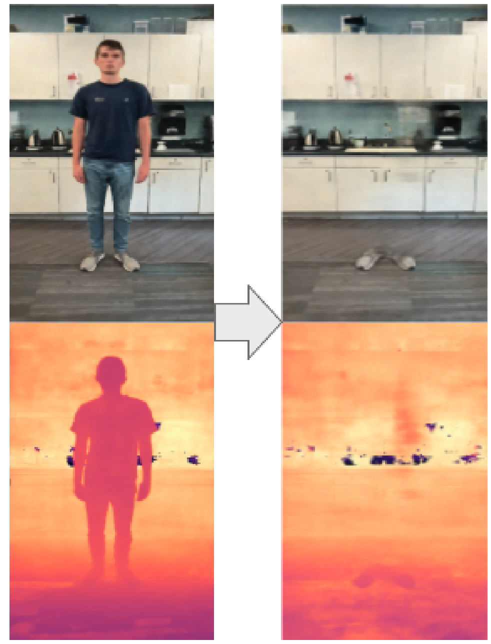


Fig. 6. When applied to scenes in which the object is automatically annotated by Body Pix 2.0, we see similar performance. The standard NeRF rendering (left) compared to the modified rendering (right) show that the person is almost entirely removed from the scene. The shoes of the person partially remain. This can be explained by imperfect annotations.

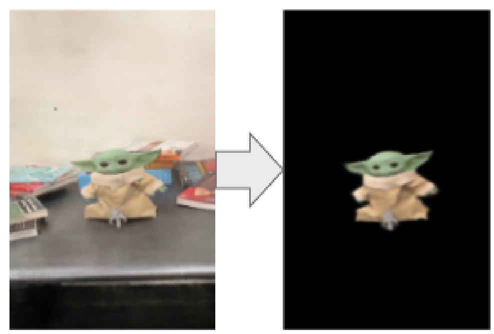


Fig. 7. By inverting the manually annotated binary mask, we can select and remove the background of the scene as opposed to the toy.

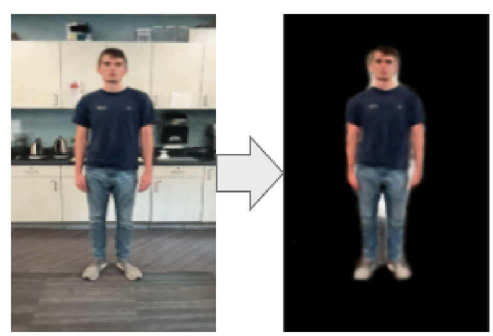


Fig. 8. Inverting the automatically annotated binary mask lets us select and remove the background of the scene as opposed to the person.

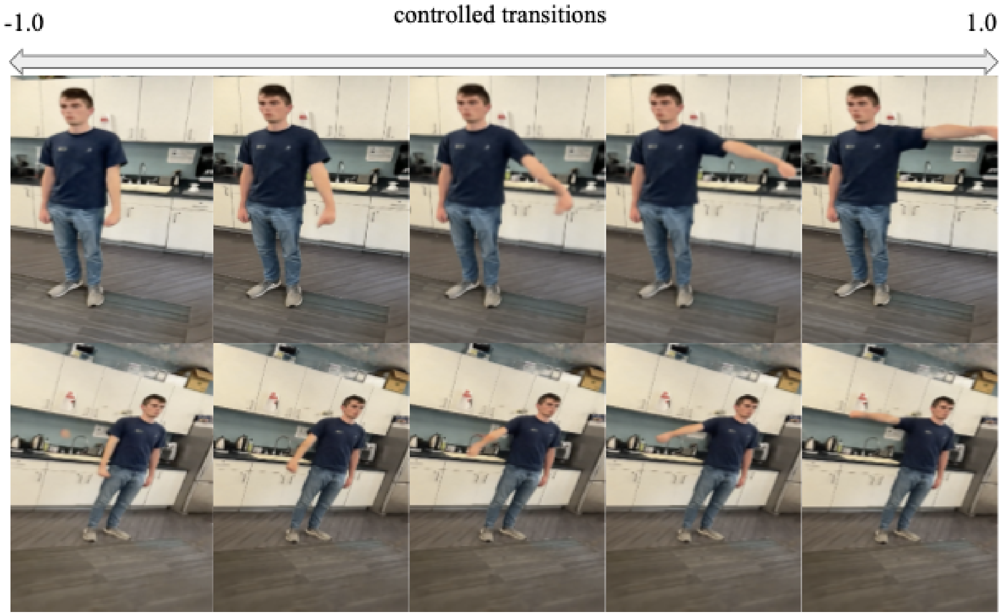


Fig. 9. Controlled output obtained from CoNeRF using automatic annotation for the arm from transition states -1.0-1.0 where -1.0 is the start state and 1.0 is the final state.

scenes, a minor modification was needed. The original CoNeRF code only accepts manual annotations generated from labelme [10] in the form of a json file and *then* converts those coordinates to a binary mask. To make this code suitable for our automatically generated masks, we simply removed the json conversion and directly used our binary masks of the dynamic component. With CoNeRF accepting our binary masks, we now supplied transition values, from -1 - 1, to a sparse set of the captured frames. The final controllable output for this experiment of both right arm and left arm movement can be found in Fig. 9.

V. CONCLUSIONS

This work proposed a method for automated content editing in NeRFs that allows for simple manipulation of both static and dynamic neural radiance fields. Using the automatically generated masks from Body Pix 2.0 and our ray removal method, static scenes can be rendered without a person present and the background intact or without the background and the person unaffected. When applying our use of automatic annotation to preexisting dynamic NeRF methods, such as CoNeRF, we can remove the need for manual annotation of the controllable component. Both intended applications of this method proves to be successful.

While the experiments presented within this paper were a success, there still remains numerous directions for future work. A few of the most promising steps are to improve the masking coverage to more fully capture the object of interest, apply different segmentation methods to automatically annotate objects of different classes, extend this method to other dynamic NeRF variants that rely on manual annotation, and increase the editing possibilities.

ACKNOWLEDGMENT

This project was funded by the National Science Foundation (NSF) under Grant No. 1659774. I want to say thank you to the Robotics Institute Summer Scholar program organizers Dr. John M. Dolan and Rachel Burcin, my faculty mentor Dr. László A. Jeni, and Dr. Simon Lucey for the many helpful conversations. I will forever be grateful for this engaging academic and social experience.

REFERENCES

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *CoRR*, vol. abs/2003.08934, 2020. [Online]. Available: https://arxiv.org/abs/2003.08934

[2] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," 2022. [Online]. Available: https://arxiv.org/abs/2203.09517

[3] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5501–5510.

[4] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, and U. Neumann, "Point-nerf: Point-based neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5438–5448.

[5] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "Nerf++: Analyzing and improving neural radiance fields," *arXiv preprint arXiv:2010.07492*, 2020.

[6] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhöfer, C. Lassner, and C. Theobalt, "Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a deforming scene from monocular video," *CoRR*, vol. abs/2012.12247, 2020. [Online]. Available: https://arxiv.org/abs/2012.12247

[7] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10 313–10 322.

[8] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz, "Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields," *arXiv preprint arXiv:2106.13228*, 2021.

[9] K. Kania, K. M. Yi, M. Kowalski, T. Trzcinski, and A. Tagliasacchi, "Conerf: Controllable neural radiance fields," *CoRR*, vol. abs/2112.01983, 2021. [Online]. Available: https://arxiv.org/abs/2112.01983

[10] K. Wada, "labelme: Image polygonal annotation with python," https://github.com/wkentaro/labelme, 2018.

[11] J. T. Kajiya and B. P. Von Herzen, "Ray tracing volume densities," *ACM SIGGRAPH computer graphics*, vol. 18, no. 3, pp. 165–174, 1984.

[12] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Deformable neural radiance fields," *CoRR*, vol. abs/2011.12948, 2020. [Online]. Available: https://arxiv.org/abs/2011.12948

Foldable Compliant Origami Swimmer

Kaylie Alexandra Barber[1*], Kylie Marie Barber[1*], and Zeynep Temel[2]

***Abstract*— Swimming is a useful form of locomotion to achieve in robotics for exploration in inaccessible underwater locations. This paper presents an underwater swimmer that uses origami-based mechanisms to achieve its locomotion. Origami mechanisms allow for rapid, low-cost, two-dimensional manufacturing, compact storage, and the use of compliant materials that can withstand and adapt to environmental changes and pressure. Strategic placement of folds within the mechanisms allow the design to be structured for specific applications. Thus, origami is a useful tool in creating underwater robots. In this paper, we present three origami mechanisms designed through lamination procedures that demonstrate propulsion in water. One of these mechanisms, which combines the other two mechanisms, is developed into an underwater swimmer which measures 160 mm by 260 mm by 65 mm with 130 mm propulsion fins and weighs 105 g with the actuator. It is actuated in water to demonstrate its propulsion capabilities and reaches a linear speed of of 2.1 cm/s. This mechanism shows potential for future control of both direction and depth with improved actuation systems and constraints.**

***Index Terms*— Mechanism Design, Compliant Joints and Mechanisms, Biologically-Inspired Robots, Kinematics, Marine Robotics**

I. INTRODUCTION

A. Swimming

Swimming is an abundant form of locomotion in nature with many creatures exhibiting unique modes of swimming locomotion. Sea scallops, create thrust through jet propulsion, quickly opening and closing their two shells to create a jet of water to propel themselves [1]. Frogs exhibit drag-based propulsion by using their webbed feet to stroke their legs opposite their desired direction of motion, which pushes them forwards [2]. Most fish use either undulatory motion, oscillatory motion, or a combination of these motions to achieve propulsion. While eels move their flexible bodies back and forth in undulatory motion, boxfish have more rigid bodies and generate thrust through their fins [3].

All these types of swimming break symmetry in some way. There are two different types of symmetry that can be broken to allow for swimming: time symmetry or shape symmetry. Breaking time symmetry entails moving through a motion quickly in one direction and slowly in the opposite direction. Breaking shape symmetry entails the mechanism or animal changing its geometry based on the direction that it is moving [4]. Swimming robots follow these same principles of breaking symmetry. Without breaking symmetry, a swimming robot robot would oscillate, but breakage of either time or shape symmetry allows it to achieve a dominate direction of movement.

B. Origami

Our research incorporates origami into swimming robot design which will provide the benefits of soft robots with the control of rigid robots. Origami is a method of folding materials in different ways to achieve desired structures and motions. It has been employed within many engineering applications to achieve geometrical transformation and create new designs and movements, such as space applications with a deployable solar panel array [5]. In incorporation of origami into engineering designs, materials must often be changed from the traditional material of the art, i.e., papers. This can introduce challenges with thickness accommodation, where the inherent thicknesses of materials can inhibit the folding of the mechanism [5]. Furthermore, cuts, although not traditionally used in origami, may be used to simplify the mechanism and remove unnecessary sections. This broader definition of origami will be used in the design of the mechanisms within this paper.

Using rigid materials as planar components and flexible materials at the folding points, we can create origami that behaves in a compliant manner. Careful placement of these folding points allows the motion of the overall mechanism to be designed according to the desired trajectory. Folds can also be strategically placed to create multistability, allowing the mechanism to store energy and use its elasticity to actuate itself and reduce the need for complex actuation systems [6]. Lamination procedures, consisting of the alteration of rigid, adhesive, and compliant layers, can be used to create fast, low cost, and durable methods for fabrication that can be easily modified such that the design is tunable and adaptable [7]. Origami offers an efficient manufacturing process, allowing one to manufacture a mechanism from a two-dimensional state and assemble the mechanism into a three-dimensional state. The mechanism can be fixed into a three-dimensional state after fabrication, or it can be designed as deployable where it converts from a two-dimensional to three-dimensional state when actuated. [8].

II. LITERATURE REVIEW

Origami has been successfully applied to robotics due to its easy manufacturability, low cost, and compliance. Research has shown promise for using origami in robots targeted towards land environments. For instance, Peri, a

[1]Kaylie Alexandra Barber and Kylie Marie Barber are in Mechanical and Mechatronics Engineering with the Schreyer Honors College at The Pennsylvania State University.

[2]Zeynep Temel is with the Robotics Institute at Carnegie Mellon University.

[*]Equal contributors.

< TABLE OF CONTENTS

bio-inspired, pseudo-compliant crawling robot, utilizes two Kresling towers which are rigidly connected and independently actuated by servo motors such that the towers expand and contract to enable the robot to crawl forwards and turn. The Kresling tower design gives this robot a high level of flexibility to deform based upon actuation and its environment [9]. Recently, origami is being explored within aquatic environments. One origami design takes inspiration from pelican eels, using the flexibility of origami to achieve dual-morphing with derivations of Miura and Yoshimura origami patterns. When actuated, this origami-based mechanism can unfold and expand itself to another stable position [10]. In another origami-design, researchers at the University of Pennsylvania created a jet propulsion swimmer which uses dielectric elastomers to actuate a flexible polyethylene terephthalate film (PET) surrounding an acrylic tube which contains the electronics of the system. The PET film is folded into a magic ball origami waterbomb tessellation which transitions between an ellipsoidal and spherical shape when actuated. The jet swimmer weighs 620 g and can achieve a velocity of 6.7 cm/s, which is 0.2 body lengths per second. In its ellipsoidal shape, it is 30.5 cm long with a 19 cm diameter mechanism, while in its spherical shape, it is 25 cm long with a 23 cm diameter mechanism. The jet swimmer propels forward but lacks steering and depth control over its swimming [11]. Non-origami compliant swimmers have achieved propulsion, steering, and depth control [12]; however, they often require complex molding processes, which requires more steps resulting in a longer fabrication time as compared to such emerging origami robots. To the best of our knowledge, origami swimming robots have not yet offered depth control or steering despite the flexibility in design and actuation that they offer. In this paper, we demonstrate three origami mechanisms which together demonstrate propulsion. In future research, we plan to demonstrate these mechanisms' potential for depth control and steering.

III. METHODOLOGY

A. Mechanism Design

The mechanisms were developed using origami folding paper to establish base parameters and folding patterns for the designs. After such iterations, the designs were optimized in CAD software to improve motion constraints and achieve the desired ranges of motion. Once these mechanisms were designed, two-dimensional folding patterns were also drawn in CAD software for use as the laser cutting patterns during the lamination process.

YoFin, shown in Figure 1, is a bistable serial six-bar linkage actuated with a single servo motor which transfers the mechanism between its two stable positions. The servo motor pushes the mechanism to its compressed stable state as shown in Figure 1, while the elastic deformation of the hinges allows for the mechanism to spring back to its manufactured open stable state as shown in Figure 1. YoFin mechanism slowly closes and forcefully extends to cause an unequal amount of force which has the potential to propel the mechanism when released.

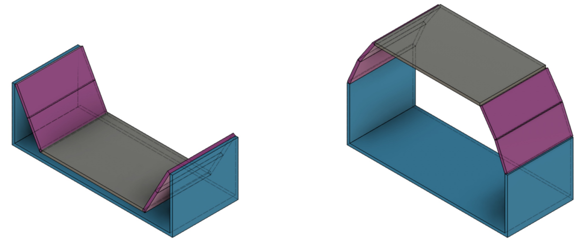


Fig. 1. Model of YoFin mechanism in compressed position and open position used for actuation in YoDiFin

DiamondFin, shown in Figure 2, is a parallel six-bar linkage which has fins that can move together or separately depending on the actuation. Just as in the YoFin design, the actuation speed can be modified to close the mechanism slowly and then allow the elastic deformation to spring the mechanism to its original state. A single servo motor can be used to pull the fins into a closed position from which they can be released as shown in Figure 2, or each fin can be actuated independently to allow for control over the direction of the resulting propulsion as shown in Figure 3.

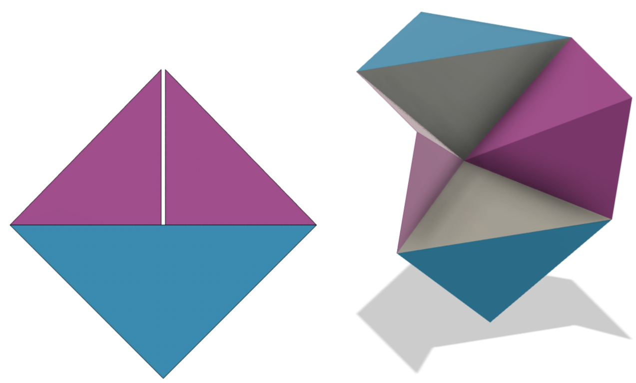


Fig. 2. Model of DiamondFin mechanism in compressed position and open position used for propulsion in YoDiFIN

These mechanisms can be designed to combine together as a single mechanisms to form the swimmer called YoDiFin shown in Figure 4. YoDiFin measures 160 mm x 260 mm x 65 mm with 130 mm propulsion fins. YoDiFin weighs 45 g without the actuation system and 105 g with the 60 g actuation system. The combination of YoFin and DiamondFin into a single mechanism allows for actuation of both mechanisms together with a single servo motor as seen in Figure 5. In Figure 5 the mechanism is in its open position before actuation of the servo motor and then in its closed position after it has been actuated.

B. Lamination

All mechanisms are manufactured using lamination, which compresses multiple materials together into one cohesive mechanism. The layers are composed of rigid, flexible, and

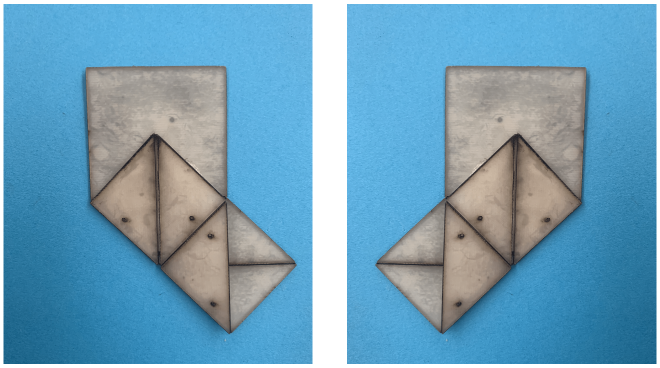


Fig. 3. DiamondFin with one fin actuated, showing future potential for direction control of mechanism

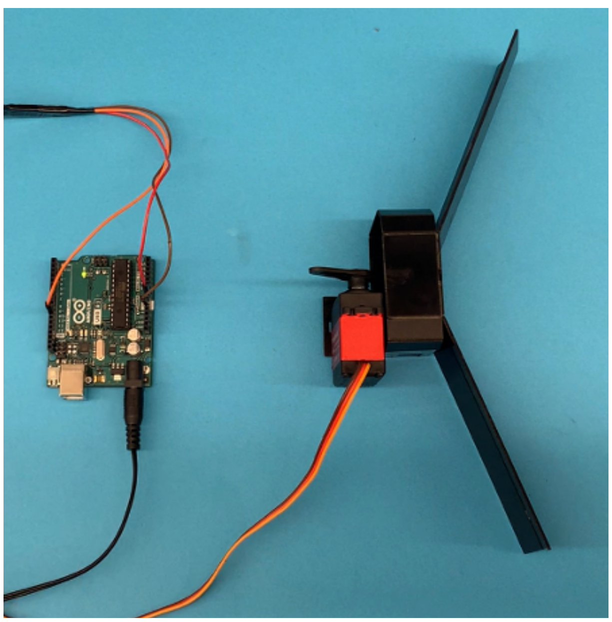


Fig. 4. Image of YoDiFin in open position with the actuation system shown

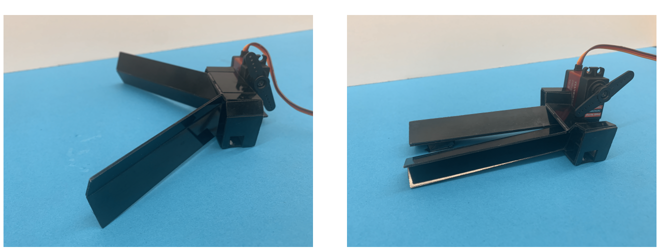


Fig. 5. Image of YoDiFin in open position and closed position after being actuated by the servo motor

adhesive materials as shown in Figure 6. The rigid layer uses 0.508 mm thick FR-4, a glass fiber composite, which provides structure to the mechanism. The flexible layer uses 0.1 mm thick polyethylene terephthalate (PET) to create deformable hinges. 0.045 mm thick 3M double-sided tape adheres these layers together. These individual layers are shown in Figure 7. Cuts are placed in the rigid and adhesive layers to create a section with only the flexible layer. This creates compliant joints where the mechanism can bend for a controllable distance and direction.

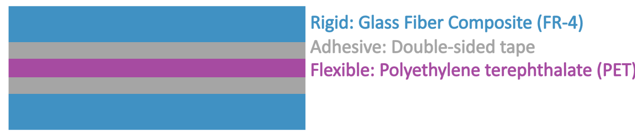


Fig. 6. Diagram of layers used in origami-lamination procedure

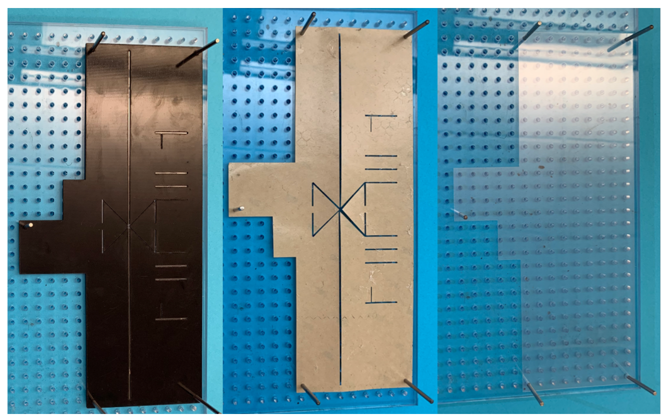


Fig. 7. Layers in the Lamination process with the rigid layer (left), adhesive layer (middle), and flexible layer (right).

Each layer was designed using CAD software and cut using a Universal laser cutter. Cuts are made in the rigid and adhesive layers to accommodate for the thickness of the material and create hinges where the adhesive material could bend for a restricted distance controlled by the width of the cuts. An exterior support is cut with each layer of the mechanism to provide area for alignment pins. This exterior support is removed once all layers are laminated together using the laser cutter to release the mechanism's joints.

Once the mechanism is released, it is assembled into its designed three-dimensional orientation. This involved folding the mechanism and securing it by adhering specified panels together using Loctite 416 adhesive and Loctite 712 accelerator. Panels are cut and opened like doors in the side panels intended to be rigid to constrain the angle to 90 degrees. These panels, shown in Figure 8, are likewise adhered to the bottom surface with the same glue and accelerator listed above.

C. Actuation

< TABLE OF CONTENTS

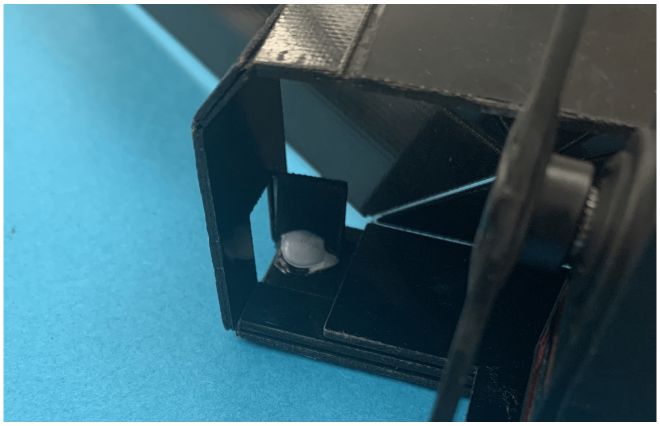


Fig. 8. Panel that acts as a door to enforce 90 degree panels

To actuate YoDiFin, we harness the linear compression of the YoFin mechanism to close the wings of the DiamondFin mechanism. This allows for a single motor to compress both fins. The servo motor used is a TIANKONGRC TD 8125MG 360 degree continuous waterproof digital servo motor which weighs 60 g. This servo motor is fitted with a double propeller attachment which presses the panel of the YoFin mechanism, causing it to move to its compressed position and thereby closing the fins of the connected DiamondFin mechanism. Once the mechanism has been depressed, the propeller on the motor slips off of the panel, releasing the mechanism. The elasticity of the mechanism through the bistable designs causes the mechanism to return to its open, manufactured state.

D. Testing

To test the YoDiFin mechanism, the servo is fitted with a double propeller attachment and glued to the motor support panel. The lowest servo speed is used for actuation. Averaging the times to reach 10 turns on the servo yields an average angular speed of 0.6 revolutions per second. Since a double propeller attachment is used, the mechanism would be compressed 1.2 times each second under ideal conditions.

The tank used for the experiment is a 51.12 cm x 25.72 cm x 31.75 cm 37.85 liter glass aquarium full of water. For flotation, pieces of foam are glued to specific areas of the mechanism which had the most mass as determined by analyzing how the robot sunk and tipped when placed in the water without flotation devices. Foam is attached at the base of the wings as well as on the motor. The tether of three wires for controlling the servo motor with the external Arduino rests in the water so that the robot moves freely without support from the wires hanging over the edge of the tank as shown in Figure 9.

Once setup is complete, the servo actuates the robot so that it swims across the tank. Video footage is taken of the robot during its progression and used to determine the time taken to swim across the tank.

IV. Results & Discussion

A. Propulsion

Linear swimming speed for the mechanism is calculated using the distance the robot swam as well as the time it took the robot to swim the distance. Our trial yields a swimming speed of 2.1 cm/s with the actively-controlled closing of the fins generating the propulsion. However, the swimming direction is counter-intuitive to our expectations and design. The actuation system is designed as a slow close actuation and a fast open passive elastic bistable snap to the original position. Therefore, we expected that the robot would move by using its fins to pull itself through the water. However, the closing of the fins generates a water jet which is stronger than the snap of the fins, causing the robot to swim in the backwards direction. With adjustments to the timing and actuation system, we believe that the robot's speed will increase and further that there may be potential to move both forwards and backwards by alternating between a fast and slow fin close.

B. Actuation

During the process of actuating the robot within the water, the robot started to delaminate, with the FR-4 rigid material peeling away from the flexible layer as shown in Figure 10. We believe that this was caused due to the torsion imposed on the system during actuation. As the propeller slides off of the top panel, it causes that panel to twist, causing torsional forces that pull the layers apart. To fix this issue, we suggest either laminating using a stronger adhesive or devising a new contact surface between the actuation system and the YoDiFin mechanism that would reduce the twist as the propeller slides off to initiate the snap. The delamination caused severe alterations to the dynamics of the system, preventing further speed tests from being performed to better approximate the swimmer's speed. Furthermore, as the propeller applies force to begin to actuate the mechanism, the support panel that attaches it to the mechanism flexes, causing the motor to not be able to fully compress the mechanism. We propose using a more rigid material to keep this panel stable, allowing the servo motor to fully actuate the YoDiFin mechanism. With full actuation, this could also change the swimming direction of the mechanism by increasing the distance that the bistable snap acts over.

V. Future Work

Because the lamination process is manual, manufacturing errors can be easily introduced into the design during fabrication. Therefore, we would like to enforce additional motion constraints into the design of the mechanism in order to further constrain the mechanism to the desired degrees of freedom. We would further like to experiment with different rigid and adhesive materials since the rigid material is bending during actuation and the resulting torsional forces are causing damaging delamination to the mechanism. Incorporating a more rigid outer layer and stronger adhesive will help protect the mechanism from this delamination.

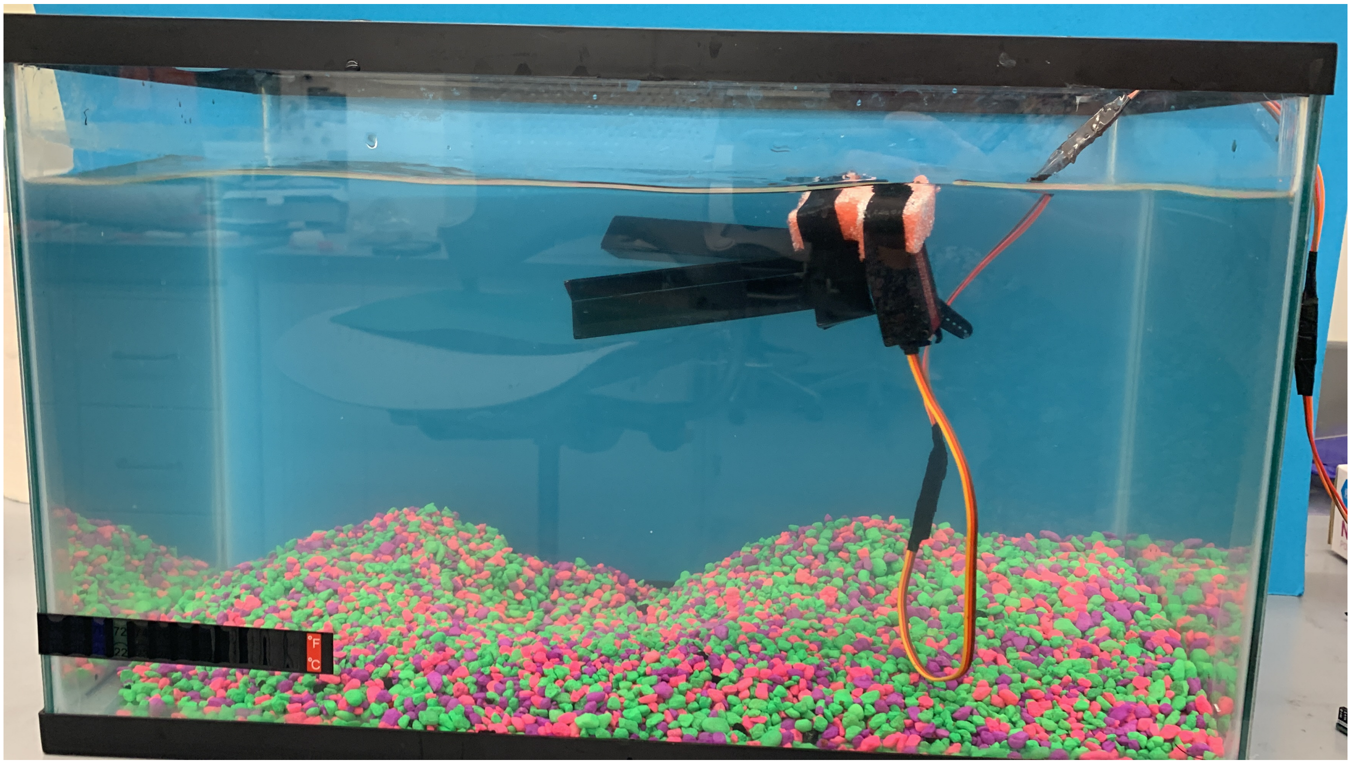


Fig. 9. Testing setup with YoDiFin inside water-filled aquarium

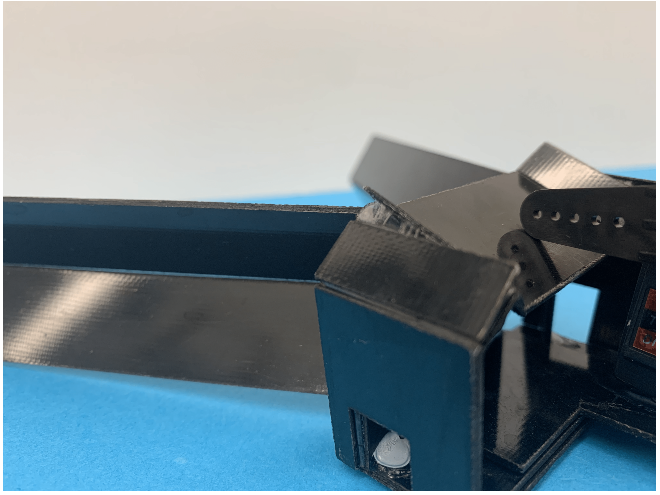


Fig. 10. Delamination of YoDiFin panel

To better understand the propulsion direction of YoDiFin, we would like to generate a mathematical model or simulation to compare the different pressures and forces acting on the mechanisms in order to incorporate better control and optimize the geometry. Furthermore, since the designed mechanisms show potential for further motion control with respect to depth and direction, we would also like to explore these aspects with other design iterations and actuation systems. Control of the mechanisms can also be improved through the incorporation of sensors to help guide the system and make it autonomous. Making these design changes and further characterizing the swimmer will allow us to apply it in underwater discovery and better understand how to use origami, lamination, and compliant mechanisms in robotics.

VI. Conclusions

YoDiFin exhibits successful use of origami to propel forwards in water. It uses a simple actuation system and two bistable compliant origami mechanisms to create this motion. Though demonstrated in propulsion only, the individual DiamondFin and YoFin mechanisms show promise for both direction and depth control respectively. YoDiFin can be used for exploration and education to demonstrate origami in engineering, the lamination process, fluid dynamics, and swimming methods.

Acknowledgment

The authors would like to thank the Robotics Institute Summer Scholars (RISS) program and Carnegie Mellon University for the resources and guidance which allowed them to conduct this research. They would further like to thank the National Science Foundation (NSF) for providing funding for this project.

References

[1] G. Mason, J. Sameoto, and A. Metaxas, "In situ swimming characteristics of the sea scallop, placopecten magellanicus, on german bank, gulf of maine," vol. 94, p. 1019.

[2] F. Jizhuang, Z. Wei, Y. Bowen, and L. Gangfeng, "Propulsive efficiency of frog swimming with different feet and swimming patterns," vol. 6, no. 4, pp. 503–510. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5399553/

[3] Y. Zhong, Z. Li, and R. Du, "A novel robot fish with wire-driven active body and compliant tail," vol. 22, no. 4, pp. 1633–1643, conference Name: IEEE/ASME Transactions on Mechatronics.

[4] A. Cho. Tiny symmetric swimmer evades basic rule of fluid dynamics. [Online]. Available: https://www.science.org/content/article/tiny-symmetric-swimmer-evades-basic-rule-fluid-dynamics

[5] S. A. Zirbel, R. J. Lang, M. W. Thomson, D. A. Sigel, P. E. Walkemeyer, B. P. Trease, S. P. Magleby, and L. L. Howell, "Accommodating thickness in origami-based deployable arrays1," vol. 135, no. 11. [Online]. Available: https://doi.org/10.1115/1.4025372

[6] K. Saito, A. Tsukahara, and Y. Okabe, "Designing of self-deploying origami structures using geometrically misaligned crease patterns," vol. 472, no. 2185, p. 20150235, publisher: Royal Society. [Online]. Available: https://royalsocietypublishing.org/doi/10.1098/rspa.2015.0235

[7] D. M. Aukes, B. Goldberg, M. R. Cutkosky, and R. J. Wood, "An analytic framework for developing inherently-manufacturable pop-up laminate devices," vol. 23, no. 9, p. 094013, publisher: IOP Publishing. [Online]. Available: https://doi.org/10.1088/0964-1726/23/9/094013

[8] M. Meloni, J. Cai, Q. Zhang, D. Sang-Hoon Lee, M. Li, R. Ma, T. E. Parashkevov, and J. Feng, "Engineering origami: A comprehensive review of recent applications, design methods, and tools," vol. 8, no. 13, p. 2000636, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/advs.202000636. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/advs.202000636

[9] K. Gustafson, O. Angatkina, and A. Wissa, "Model-based design of a multistable origami-enabled crawling robot," vol. 29, no. 1, p. 015013, publisher: IOP Publishing. [Online]. Available: https://doi.org/10.1088/1361-665x/ab52c5

[10] W. Kim, J. Byun, J.-K. Kim, W.-Y. Choi, K. Jakobsen, J. Jakobsen, D.-Y. Lee, and K.-J. Cho, "Bioinspired dual-morphing stretchable origami," vol. 4, no. 36, p. eaay3493. [Online]. Available: https://www.science.org/doi/10.1126/scirobotics.aay3493

[11] Z. Yang, D. Chen, D. J. Levine, and C. Sung, "Origami-inspired robot that swims via jet propulsion," vol. 6, no. 4, pp. 7145–7152, conference Name: IEEE Robotics and Automation Letters.

[12] G. Li, X. Chen, F. Zhou, Y. Liang, Y. Xiao, X. Cao, Z. Zhang, M. Zhang, B. Wu, S. Yin, Y. Xu, H. Fan, Z. Chen, W. Song, W. Yang, B. Pan, J. Hou, W. Zou, S. He, X. Yang, G. Mao, Z. Jia, H. Zhou, T. Li, S. Qu, Z. Xu, Z. Huang, Y. Luo, T. Xie, J. Gu, S. Zhu, and W. Yang, "Self-powered soft robot in the mariana trench," vol. 591, no. 7848, pp. 66–71, number: 7848 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41586-020-03153-z

[13] W.-S. Chu, K.-T. Lee, S.-H. Song, M.-W. Han, J.-Y. Lee, H.-S. Kim, M.-S. Kim, Y.-J. Park, K.-J. Cho, and S.-H. Ahn, "Review of biomimetic underwater robots using smart actuators," vol. 13, no. 7, pp. 1281–1292. [Online]. Available: http://link.springer.com/10.1007/s12541-012-0171-7

Benchmarking of AI-Driven Predictive Maintenance (PMx)

Maja Margul, Kyle Miller Ph.D., Robert Edman Ph.D., Merritt Kowaleski, Artur Dubrawski Ph.D.

Abstract— **Predictive maintenance (PMx) forecasts the future performance of a machinery component, given its historical feature data. It encompasses a variety of topics, including but not limited to: failure prediction, failure diagnosis (root cause analysis), failure detection, failure type classification, and recommendation of mitigation or maintenance actions after failure. When predictive maintenance is done properly, it can help reduce repair costs, cut production downtime, and enhance operational safety. Conducting a PMx analysis on gathered data allows us to identify patterns that lead to potential problems or failures. We can use this detailed information to fix problems before they happen, which allows us to optimize equipment lifetime and minimize downtime. However, there are some practical challenges that impede progress in PMx - one of them is access to datasets which could be broadly available for benchmarking AI-driven PMx. Most relevant datasets are considered sensitive by their owners and are therefore very difficult to obtain. Because of that, our main motivation is to provide sources and data that could be used in the PMx field. The goal is to research and assemble literature relevant to the applications of PMx in various fields - particularly literature with supporting data - and also provide summaries of those papers. These summaries include essential information about how to use the dataset to build the predictive model, and provide the results received from the experiments conducted. All the findings would be summarized in a research paper. The final step is to create a repository on GitHub containing a large collection of benchmark datasets found on the internet. Based on our work, people all over the world would be able to get information about PMx and related datasets with sample algorithms given for analysis purposes.**

Index Terms— **Machine learning, Predictive Maintenance, Deep Learning Methods**

I. INTRODUCTION

Failures cause costly and extremely troublesome disruptions to production processes. By predicting failures, it is possible to take preventive measures in advance, thus avoiding failures and minimizing their consequences. Predictive Maintenance is based on the collection of data about the condition of machines and the course of their processes. Based on this data, models are created to predict the occurrence of failures and determine the condition of the equipment, such as the degree of wear. The models are applied on an ongoing basis as processes are running to determine in real time the risk of failure and alert for possible problems. By implementing a predictive maintenance strategy, failure and service costs can be reduced, equipment effectiveness can be improved, downtime can be reduced, and up-time and safety can be increased. Taking into consideration different types of Predictive Maintenance models, we can distinguish 2 major categories:

- data-driven (models using machine learning)
- physics-based models.

Most often, physics-based models are harder because they require very advanced knowledge about the particular domain, while data-driven models attempt to create a predictive model automatically from the given data.

II. How does Predictive Maintenance work?

Predictive Maintenance (PMx) is a form of a proactive maintenance that uses real-time sensor data, historical performance data and advanced analytics to predict when asset failure will occur. In other words, it is the analysis of data coming directly from machines in order to predict potential failures and take maintenance actions before the failures occur. Predictive maintenance can significantly reduce costs directly related to servicing machines, as well as additional costs resulting from unplanned downtime.

Predictive Maintenance is based primarily on collecting very large amounts of data provided directly from sensors mounted on machines and also from the shop floor/hangar, and then analyzing the resulting data to catch anomalies that may indicate a risk of failure. Parameters that can be measured include temperature, machine/component vibration levels, oil consumption, and pressure. A very useful field to help collect large amounts of data and analyze them is IoT (Internet of Things). The use of tools from this field gives the opportunity to monitor the production process, machine operation, and production conditions in real time. Data streams from various types of sensors are collected to be further analyzed using machine learning models, which makes it possible to predict with high accuracy the risk of failure. In addition, having historical data and ongoing analysis of failures makes it possible to continuously improve the reliability of models and predict failures with greater efficiency.

III. Choosing an effective machine learning technique for predictive maintenance

In the field of predictive maintenance, various machine learning techniques are used to learn from both historical and live data and perform analysis of different failure patterns. Thus, it is important to select a machine learning technique that will produce the best possible results for the project. The first step is to collect sufficient and high-quality time-series data (to this end, one of the tasks of our project was to create a repository that stores various databases that are used in predictive maintenance analysis. A description of this repository can be found in Chapter 5 of this article). Each machine to be studied must be continuously monitored and its sensor data collected. This data is then segregated to make visible the factors affecting the machine's operation. These factors vary widely and depend on the specific equipment type. For example, in the case of engines, such factors could be temperature, humidity, oil level and density, vibration, etc. The process of locating relevant factors affecting the operation of a machine is as follows: First, identify the types of failures that are possible. Then, analyze the parts, components and processes that may contribute to these types of failure. Finally, determine which parameters will provide useful information on the relevant processes occurring in the machine. In order to obtain enough data to make a detailed analysis, it is best to conduct monitoring and data collection for several years. Once the data has been properly prepared for analysis, the next step is to select the appropriate machine learning technique. To this end, the first consideration is to determine the type of output that the predictive model should give. It is important to record, tag and identify all events occurring, which will allow for quick filtering of the data. Going forward, this will provide an opportunity to identify indicators/sensors that are having a favorable impact on the machine and those that may indicate a progressive failure. It may also be helpful to determine if there is any potential relationship between the number of events occurring and the number of failures occurring, and to determine if there is an event that occurs only before the failure occurs and never under other circumstances. In addition, it may be useful to determine the minimum time period required to signal potential failures and hazards. With such information, it is possible to decide what technique to adopt for creating a predictive model in order to best fit the type of database being used. For this purpose, we will consider 5 main machine learning techniques:

1) Regression Models to Predict Remaining Useful Life (RUL) - Both static data (such as the date the equipment was made, the model, the start date of service, and the location of the system) and historical data (i.e., sensor data) are necessary here. It is necessary to mark and record every event and failure that occurred. Such information is used to train the model to predict possible failures. At first, consider a scenario in which the model will be focused only on one type of failure. Then we are dealing with a gradual degradation process. When the model is to consider different types of failures, then this process proceeds differently, which can affect the accuracy of the results. For this reason, it is recommended that a different model be adopted for each type of failure. The result of this technique is a model that provides output in the form of the number of days left before a failure occurs, or Remaining Useful Life (RUL).
2) Regression Models to Predict Degradation State - This is similar to RUL prediction, but the target variable is something more directly representing wear, such as mm of wear on a cutting die, as opposed to a number of hours or days. Degradation state can inform RUL predictions, but it is a feature that can be confirmed immediately rather than a forecasting problem.
3) Classification Model To Predict Failure Within a Predecided Time Frame - It can be extremely difficult to create a model capable of accurately analyzing the full life of a machine. However, often the maintenance team only needs to know whether there is a risk that the machine will fail in the near future. For this, it is best to use a classification model to predict whether a machine will fail within N days or cycles. Similarly to RUL prediction, such a model requires both static and historical data. Also similarly to RUL prediction, you must properly characterize, attribute and label the events that occurred during the machine's operation. This classification technique, unlike a regression model, does not necessarily assume gradual degradation of the machine, due to the fact that here we do not predict the exact time, but only look for an appropriate time frame. Therefore, classification models provide an opportunity to account for multiple types of failure within a single model (for this purpose, the model should be considered a multi-class problem). In order to build such a model, it is necessary to have data with appropriate labels assigned to different types of failure, and a sufficient number of instances of each type of failure occurring in order to properly train the model.
4) Flagging Anomalous Behaviour - This technique proves extremely useful when dealing with mission-critical systems, where the number of failure events is limited. This means that there will be a limited number of failures that the team is able to analyze in order to build a model. Not all incidents are tagged, recorded or otherwise available. The goal is to use data to identify normal behavior and distinguish it from anomalous behavior which can lead to failure. Can be thought of as a semi-supervised/unsupervised version of failure classification.
5) Fault Diagnosis / Root Cause Analysis - Similar to failure classification, but instead of predicting failure within a certain time frame, the model classifies the root cause of a failure that has already occurred. It may also provide recommendations of corrective actions to take.

In conclusion, the model should be selected depending on the data you have and the type of output you want. The most important thing is to initially understand the data, the problem and the conditions under which the machine operates in order to enable correct analysis and desired results. This chapter was written based on the information contained in this article.

IV. Case Studies of Literature About Applications of Predictive Maintenance

This chapter summarizes a number of different articles on predictive maintenance applied in various scientific fields. The purpose of this research was to analyze various experiments in terms of the problem that was studied, the models used, and the organizational context under which the models were built and developed.

We selected 10 research papers (some of them use datasets from the AutonLab PMx GitHub Repository, which is described in the last section of this paper). The titles are as follows:

1) Predictive Maintenance, its Implementation and Latest Trends [1]
2) Predictive Maintenance for General Aviation Using Convolutional Transformers [2]
3) On Construction of Early Warning Systems for Predictive Maintenance in Aerospace Industry [3]
4) Predictive Maintenance for Aircraft Components Using Proportional Hazard Models [4]
5) Predictive Maintenance in Aviation Failure Prediction from Post-Flight Reports [5]
6) A Research Study on Unsupervised Machine Learning Algorithms for Early Fault Detection in Predictive Maintenance [6]
7) IoT-Based Predictive Maintenance in Manufacturing Sector [7]
8) Investigating Strategies and Parameters to Predict Maintenance of an Elevator System [8]
9) A Comparative Study of State-of-the-Art Machine Learning Algorithms for Predictive Maintenance [9]
10) Prediction of Failures in the Air Pressure System of Scania Trucks using a Random Forest and Feature Engineering [10]

Each of the above articles delves into the essence of predictive maintenance in detail. We will try to bring out the most important elements covered.

A. Case study of: "Predictive Maintenance, its Implementation and Latest Trends"

This research paper, authored by Sule Selcuk [1], examines various methods, techniques and trends used in PMx and offers suggestions for the implementation of predictive maintenance programs in factories, among others. The author emphasizes in his article the essence of such programs, which help to detect early signs of damage/failure and then initiate maintenance procedures in a timely manner. The paper classifies and describes 5 Predictive Maintenance techniques: process parameter measurements, vibration analysis, oil analysis, thermal analysis, and acoustic analysis.

The first technique described is process parameter measurements, where parameters such as process efficiency, heat loss, temperature, motor current, fluid pressure, humidity and flow rate are measured, among others. Relevant here are any abnormal changes in the values of these parameters, changes in the rate of production as well as product quality. All these aspects can provide information about the health of the system.

The next technique described is vibration analysis, which is also one of the most popular techniques used in predictive maintenance. It is most often used in rotating or reciprocating equipment to obtain information about the state of the system. Quite often, emerging and increasing vibrations can indicate the progressive failure of the components in question, and due to the difficulty of interpretation, the use of artificial neural networks in this technique often comes to the rescue. For example, by having a benchmark of a healthy system as a reference and then comparing it to other benchmarks, it is possible to detect and identify problems of the component under investigation.

Another technique described by the researcher is oil analysis, which uses data on the oils and lubricants used to obtain valuable information about the condition of the machine, as well as the condition of the oil or lubricant itself. This technique can be divided into 2 types: particle wear analysis, which is used to verify the condition of mechanical components, and oil analysis, which is used to verify that the lubricant has not degraded. These two types of analysis can provide important information on the condition of the system and are particularly valuable for analyses performed on diesel engines, due to the opportunity to identify defects in the rotating components that make up the engine structure.

Another technique is thermal analysis (thermography), which uses the relationship between temperature and the wavelength of light to make temperature changes visible. It is a widely used technique to detect mechanical or electrical problems in which there are some temperature anomalies. Thermography can be used to study the relative or absolute temperature of a given system.

The last technique described is acoustic analysis. In this technique, sound is measured - if there are any changes in sound from the recorded reference sound, this indicates wear or deterioration of components. Sound can provide important information about the health of the system.

Another topic covered by the article's author is the Computerized Maintenance Management System (CMMS). This is software that integrates and manages information related to activities that concern maintenance (all kinds of data collection, data processing, decision-making, maintenance planning, control procedures, reporting and much more). In addition, the system monitors the backlog, determine priorities, and plans decisions effectively. Several different computational tools such as a knowledge base, neural networks or logic and fuzzy networks have been developed for decision-making.

Regarding the implementation of Predictive Maintenance, the author of the article has touched on important information that is crucial to the correct operation of PMx-related programs. Not only should attention be paid to selecting the most important components to monitor, but it is also important to specify the parameters that would indicate deterioration of the components under study, select the appropriate predictive maintenance technique, locate the sensors and set a critical threshold value for each sensor, and select the appropriate CMMS to manage the maintenance program.

The last aspect that the author addressed was research areas that support predictive maintenance work. These include E-maintenance, tele-maintenance, Internet of Things(IoT), and radio-frequency identification(RFID). Each of these areas in its own unique way makes predictive maintenance a more efficient, cost-effective and acceptable field for industries of all kinds.

Undoubtedly, the author has ardently presented the important aspects of the field of predictive maintenance, which make it one of the most important elements of the work of many plants and factories. The paper describes how PMx contributes to the reliability of the operation of many machines and components, improving safety and productivity while reducing unnecessary costs, such as those associated with unplanned maintenance.

B. Case study of: "Predictive Maintenance for General Aviation Using Convolutional Transformers"

This research paper is written by Hong Yang, Aidan LaBella, and Travis Desell [2]. These researchers conducted a detailed analysis using the database that comes from the NGAFID-MC (National General Aviation Flight Information Database, Maintenance Classification). This is a novel benchmark in terms of difficulty, number of samples and sequence length, and can be accessed in the GitHub repository described in section V. This dataset consists of over 7500 labeled flights of Cessna 172S, representing over 11500 hours of per-second flight data recorded from 23 sensors. The sensors used measure engine RPM, oil temperature, oil pressure, gasket temperature, airspeed, pitch, roll, and outside air temperature.

The main problem the researchers addressed was classifying whether flights are problematic (i.e., those that resulted in forced and unscheduled maintenance) or non-problematic (i.e., post maintenance). The essence of the task was to be able to detect those features that are relevant to classification.

The following models were selected to conduct the analysis:

1) Conv-MHSA (Convolutional Multi-Headed Self Attention),
2) two CLSTM models (Convolutional Long Short Term Memory Networks) – the first one is Conv-LSTM and the second one is EX-Conv-LSTM,
3) VAE-Conv GRU (Convolutional GRU Variational Auto Encoders).

Each of them was characterized by certain distinctive features. For example, in the context of applying algorithms to MTS (Multivariate Time Series), the MHSA model has a lot of pros that gives it an advantage over LSTM models. In particular, the MSHA model has demonstrated the ability to model long-term relationships in time series data. In this experiment, the researchers used Conv-MHSA to reduce the temporal resolution from 4096 to 512. Using Conv-LSTM, on the other hand, allows for extracting the feature from the sequence before the LSTM layers and reducing MTS temporal resolution.

The last model, VAE-Conv-GRU, which uses convolutions to reduce the temporal resolution, has been chosen for anomaly detection in MTS data. After training the models, the researchers observed that training Conv-MHSA has some significant computational advantages over all other models.

When evaluating using area under the curve score for Precision-Recall (PR) and Receiver Operating Characteristic (ROC), VAE-Conv-GRU had the worst performance, while the Conv-MHSA models perform better than the Conv-LSTM models. Moreover, The VAE-Conv-GRU Model is unable to predict pre or post maintenance. The researchers concluded that Conv-MHSA models perform much better than Conv-LSTM models on this dataset. Moreover, some attention has also been paid to augmentation; 3 extensions (cutout, mixup and cutmix) should be evaluated against different MTS augmentation methods and models. To summarize the researchers' achievements, the project provides an opportunity for NGAFID to not only give the user access to time-series data and perform various analyses, but also calculate and display the probability that a flight will require maintenance. This article has provided a lot of valuable information about NGAFID and the application of various algorithms to MTS. In addition, it proved that differentiating between pre and post maintenance flights can provide a significant benefit to the domain of general aviation.

C. Case study of: "On Construction of Early Warning Systems for Predictive Maintenance in Aerospace Industry"

Predictive maintenance and PHM (Predictive Health Management) play a key role in ensuring reliability and efficiency in the aviation industry. PHM methods provide an opportunity to predict and prevent possible failures, help reduce maintenance costs and increase fleet utilization. There are, of course, many problems to be faced when building a properly functioning and trustworthy predictive maintenance model.

In this chapter, attention will be paid to an article by E. V. Burnaev [3]. The problem the author faced is constructing a predictive model for early warning systems for diagnostics maintenance in the aerospace industry. The author aimed to predict the occurrence of rare failures based on the use of a new methodology that takes into account the properties of different technical systems and the specific requirements of applications.

The experiment aimed to develop algorithms for predicting the types of engine failures of passenger aircraft. For the experiment, the researcher used telemetry data from an A380 passenger aircraft, in which certain defects are present, changing the statistical properties of the data. These can be detected by using anomaly detection methods. Therefore, in the experiment, the author decided to include the following steps in the analysis of telemetry data to build early warning systems for predictive maintenance:

1) Identification of Subsystems - based on the detailed description of the data structure and measured parameters, clustering was carried out, which divided the feature space into groups of parameters related to the engine and a group of parameters related to the entire aircraft structure.
2) Detection of Anomalies – different methods are used to uncover structural changes in dependence patterns, simple extreme values, or any other abnormality.

3) Alarm construction – combining simple series of binary anomalies into composite alarm signals.
4) Event Matching – 2 possible approaches can be used to identify signals which are precursors to failure or warning events. These events mostly do not occur, and when they do, these signals happen a short time prior to such events.
5) Warning signal synthesis - related to possible feature failures, constructed by pooling relevant joint anomalies together.
6) Leave-one-object-out validation – this has goal of testing the stability of predictive patterns of anomalies extracted from the data. It is achieved by estimating the necessary parameters for the selected anomaly sources on the pooled sample of all but one object, and then test running the resulting early warning system on the left-out object.

Based on the mentioned methods, a detailed analysis of telemetry data from the Airbus A380 was carried out. It is worth noting that the constructed early warning models are based on a combination of several "simple" models; these either detect anomalies in the behavior of parameters from the aircraft's engine group, or parameters related to the entire aircraft structure. This provides an opportunity to determine which changes in parameters should cause an alarm. This enables maintenance engineers to find the causes of failures more quickly. It is not possible to achieve similar accuracy rates when using traditional machine learning methods (such as random forests, gradient boosting over decision trees, and neural networks) to build predictive models.

D. Case study of: "Predictive Maintenance for Aircraft Components Using Proportional Hazard Models"

The research paper summarized in this subsection was written by Wim J.C. Verhagen and Lennaert W.M. De Boer [4]. The main goal of this project was to improve the statistical reliability of assessment maintenance by incorporating the effect of operational factors. These factors were identified and assessed for their ability to reduce the number of unscheduled occurrences (like failures). Operational and maintenance real-world data was used for the experiment, which was then analyzed for potential factors that could have an impact on adverse actions that may occur during aircraft maintenance. Moreover, time-independent and time-dependent Proportional Hazard Models were applied for the purpose of generating reliability estimates. The experiment itself consists of 5 blocks:

1) Program initiation – to initiate the program and prepare for subsequent reliability and modeling steps.
2) Flight Identification – intended to help address the following problem: "the heavier the operational use of components, the higher the probability of component failure". Regarding to the mentioned problem, the task for this block is to identify flights which might have an impact on unexpected component failure.
3) Data Analysis - there are 2 different approaches used to identify the operational factors that may have an impact on the component failure: the first one is the analysis of extreme values and the second one is the analysis of the maximum difference.
4) Reliability Modeling – by standard statistical approach, the researchers apply a set of reliability models in order to analyze the component dataset. In order to determine the impact of operational factors on reliability behavior over time, 2 variants of the Proportional Hazard Model were used: Time-independent and Time-dependent proportional hazard models (PHM).
5) Future Predictions - the generation of expected failure time using reliability models established in the previous step.

In order to approach the previously mentioned problem, the researchers used 2 techniques: Extreme Value Analysis (EVA) and Maximum Difference Analysis (MDA). Each of these techniques identifies operational factors which were abnormally high during the flights and ultimately led to component failure.

The goal of the EVA technique is to narrow down the number of potentially related flights as much as possible and assign a flight to an encountered failure based on the occurrence of extreme values. To some extent, the technique can identify which operational factors were far too high. EVA analysis leads to the optimization of one flight variable at a time, while looking for optimal values in both the positive and negative directions. When optimizing in the positive direction, flights with certain observation values below the mean are "penalized" by being assigned a negative value; the same applies when optimizing flights in the negative direction, if flights with observation values above the mean are recorded. This increases the probability that the selected flights may have experienced similar extremities in operational variables.

MDA is important for time-independent proportional hazard models, that focus on mean values during a component's fail cycle. Successful execution of the EVA and MDA techniques provide an opportunity to select flights associated with failures including a limited list of operational factors most likely to cause these failures.

Nine components with under-average performance have been selected by the researchers for the experiment. The researchers' analysis of the selected components showed that in order to reduce the rate of unplanned removals, many improvements can be made to current reliability practices.

The researchers concluded that during the time modeling, most of the components were better represented by normal, log-normal, logistic Weibull and gamma distributions, as opposed to the standard exponential distribution that is currently used by the operator involved in this experiment. In addition, they concluded that reliability models of more advanced complexity with the presence of variables could reduce the number of failures occurring by 10 to 90 percent without incurring additional costs. Such an effect is achievable if one adjusts the frequency of events associated with scheduled maintenance, and also uses predicted values of variables for future operations to assess the probability of failure at the time considered. By doing so, and thus adding scheduled maintenance, there is potential to avoid unplanned failures and associated costs.

Through the reprocessed experiment described in this paper, it appears possible to identify operational factors that have a significant impact on the potential for failure. Statistical models in two versions of operational variables - time-dependent and time-independent – proved to be suitable for reliability estimation. The researchers concluded that the results of the analysis showed the tendency of the models to outperform those that are time-based in terms of accuracy.

E. Case study of: "Predictive Maintenance in Aviation: Failure Prediction from Post Flight Reports"

This subsection will summarize the article authored by Panagiotis Korvesis, Stephane Besseau, and Michalis Vazirgiannis [5]. The researchers conducted an experiment, using data collected over a period of 7 years from a fleet of 60 aircraft. The main problem they aimed to tackle is event prediction – developing a warning system that would notify aircraft engineers well in advance of impending aircraft failures. This system would provide an opportunity to guarantee a time reserve for preparing appropriate maintenance actions. Thus, this is an experiment to try to predict aircraft failures using data from post-flight reports, which can have a significant impact on the effectiveness of measures taken in the field of predictive maintenance.

For the purposes of the experiment, methods were used that perform well in predicting future failures, or hard-to-predict events that directly affect the decision for unscheduled maintenance. The chosen method for the experiment was a time-to-failure regression model, which outputs a risk function based on the present events and quantifies the risk of an upcoming failure. In developing and dual-imaging the predictive model, the researchers performed 3 methods: preprocessing, training/validation and testing/deployment. For training the random forest model and selecting the appropriate parameters, the researchers used 5-fold cross validation on the training set.

A Support Vector Machine (SVM) was used for classification. Positive and negative instances were then created. In predictive maintenance, it is important that false positives be as few as possible (even if this involves a reduction in prediction), as they result in additional, unnecessary processes that come with a certain cost and risk. Hence, the researchers decided to perform an evaluation at the episodic level, where a comparison was made between the SVM method and RFR (Random Forest Regression). The researchers concluded that RFR outperforms the SVM baseline method, which was found to have very poor performance.

As a result, avoiding false positives turned out to be impossible, which could be caused by the intervention of engineers that affected the target event, by performing actions that prevented its occurrence. Depending on the decision threshold set (the determination of which is important to obtain the best predictive results), different prediction values were obtained. In summary, in this article, the researchers presented a method for predicting future events from event logs in the context of predictive maintenance. The results clearly show that this method outperforms other commonly used baseline approaches (such as SVM). Moreover, this method has proven that it is possible to make predictions based solely on flight reports, which is a huge step forward in predictive maintenance for aviation.

F. Case study of: "A Research Study on Unsupervised Machine Learning Algorithms for Early Fault Detection in Predictive Maintenance"

This subsection will describe an article written by Nagved Amruthnath and Tarun Gupta [6]. The researchers selected a database containing data taken from a vibration sensor located in an exhaust fan, and then performed the matching of various machine learning algorithms. One of the aims for creating a project regarding fault classification is to show how important well-developed early failure detection can be; this eventually may significantly minimize catastrophic machine failures. The researchers defined fault detection as a process of identifying the abnormal behaviour of a subsystem; any deviation from standard behavior can be considered a failure. In order to analyze the problem and conduct an experiment towards fault detection using a benchmark for vibration monitoring data, the following algorithms were used: Principle Component Analysis (PCA) T2 statistic, Hierarchical clustering, K- Means clustering, C- Means, and Model-based clustering. The code written to carry out this experiment was created by the researchers using a statistical tool called R-programming.

In the course of conducting the analysis, it was discovered that when the trend line of the analyzed data approaches a certain value (in this case it was 60 observations), this indicates the occurrence of a failure. The researchers' main goal, therefore, was to conduct experiments using various algorithms to detect these failures in advance. Very important in such experiments is the proper selection of features, because if done inappropriately, the accuracy of the result will be greatly reduced. For that purpose, the first algorithm used by the researchers was PCA T2 statistic. This is an algorithm that does a great job of reducing dimensionality while preserving most of the information in the dataset. It is an algorithm that identifies patterns in the data and indicates both similarities and differences. The T2 statistic index, a measure of the variation of each sample within the PCA model, was also used. This type of statistic might be used for values measured against the threshold and any other values above it. The results received from this analysis allowed the researchers to conclude that the appearance of faults can be detected even after 41 observations. This leads to the conclusion that early detection of failures would make it easier for maintenance teams to monitor the subsequent changes and take appropriate corrective action.

The third algorithm chosen for the project is clustering analysis, which is one of the unsupervised machine learning methods. One of the known methods of that type is hierarchical clustering, which researchers decided to use for this analysis. In this method it is important to know how many clusters can be formed, because it would help to understand the different states of the data and present the data more accurately. For this purpose, the researchers used the elbow method and the nbClust package. The final result of this method was to group the data into 3 states: normal, warning and defective.

The next algorithms discussed in the experiment are K-Means clustering and C-Means. K-Means is one of the more popular unsupervised learning algorithms in machine learning. The task of this algorithm is to divide a dataset into predetermined clusters based on distance. The researchers chose to use Euclidean distance. C-Means clustering involves matching each data point to each cluster to some degree. The results obtained using these algorithms are the same as those obtained through hierarchical clustering.

The last model used was the Gaussian Mixture Model (GMM). It is commonly used when modeling data coming from groups that may differ from each other, although the Gaussian Distribution can model data points within the same group as well. In this experiment, the Gaussian finite mixture model (fitted by EM algorithm) is an iterative algorithm, where some initial random estimate starts and updates each subsequent iteration until convergence is detected. Taking into consideration everything that has been done so far, the researchers began with a hypothesis that there were 2 states in the data, healthy and unhealthy. By using PCA and T2 statistic, the researchers were able to detect faults 31 observations ahead. On the other hand, relying solely on data charts, it was only possible to observe trends 11 observations ahead, which makes the use of the two methods all the more effective. On the other hand, when unsupervised clustering algorithms were added, it yielded much more than the results obtained from the T2 statistic.

Through the use of the elbow method and the nbClust package, it was determined that the optimal number of clusters is 3. Based on the results obtained and matching the data to the corresponding hierarchical clusters, K-Means and C-means, almost the same results were obtained. 3 states were also identified: one state was healthy (due to its calibration with healthy data), the second state was warning, while the third state was defective. The final model was developed using a Gaussian finite mixture fitted model, which was fitted using the EM algorithm. The purpose of this type of model was to identify optimal clusters and classify observations into groups accordingly.

In summary, the main objective of the research was to benchmark various existing algorithms in machine learning for early error detection using unsupervised learning. Looking at the results, the researchers concluded that the T2 statistic performs better than the GMM method. The advantage of the T2 statistic is that even without domain information, it is possible to identify an error or critical condition, which cannot be achieved using clustering analysis. However, in clustering analysis, having some information about the data, it is possible to assign a healthy, warning or critical label to the clusters, which makes this method better than T2 statistics when it comes to detecting different levels of defects.

G. Case study of: "IoT based Predictive Maintenance in Manufacturing Sector"

The following subsection will discuss the article authored by Shikhil Nangia, Sandhya Makkar, and Rohail Hassan [7]. This article refers to one of the areas used in predictive maintenance: the Internet of Things (IoT). IoT sensors provide an opportunity for intelligent management in manufacturing plants by enabling autonomous information exchange, which can translate into more accurate business decisions. Given that this is quite an important topic in the field of predictive maintenance, the authors of the article decided to develop an architecture for IoT-based predictive maintenance. The project was created based on a case study from the auxiliary automotive industry, with the aim of demonstrating a model that would predict sudden failures in industrial machinery, making production and maintenance cycles intelligent. Such sensors can significantly assist manufacturing industries in predicting machine failures to enable an appropriate response before the failure occurs.

An important aspect of this article is the IoT-based Predictive Maintenance Architecture proposed by the authors. It consists of 5 components, which include:

- IoT sensors, which are used in order to monitor and further collect real-time data.
- Digital Signal, which is concerned with converting analog data to digital form in order to use it for analysis.
- Data Storage and Transfer, where previously-converted digital data is stored in a secure manner.
- Edge/Fog/Cloud Computing, where predictive maintenance algorithms are processed.
- Predictive maintenance for failure prediction - this is the last step and involves designing a predictive maintenance algorithm.

The database is derived from a proxy of automotive components, such as motors, rotors and heat exchangers, whose failure brings the entire assembly line to a halt. It is therefore important to predict the failure of these components in order to avoid such situations and thus improve product quality and save energy spent on machine work. It was decided to implement predictive maintenance techniques on heat exchangers, whose function is to cool extemporaneously in extremely high-temperature synthetic fluids flowing out of the assembly line. This was done in an effort to reduce the number of problems caused by continuous downtime due to clogged lines.

The experiment consisted of 6 phases, during which the data was properly prepared and divided into test and training sets, and then machine learning algorithms were implemented to create a predictive maintenance model. After the output data is analyzed, it is evaluated based on the predictive accuracy of the machine learning algorithms. To model the data and trigger alarms on a prediction of failure, tools such as Microsoft Azure ML Studio (Software), Tibco Statistica (Software), SAS Visual Data Mining and Machine learning (Software), Google AI platform (Software) and open-source software like R and Python can be used. For the algorithms used, the project's authors chose to use Machine Learning's (ML) binary classification, modeling the model using algorithms such as Support Vector Machine (SVM), Random Forest and Boosted Classification Trees (C&RT).

The error rate was determined by predicting the output results of all three algorithms used. If 2 of the 3 algorithms predict machine failure, the result is the occurrence of failure. The results obtained by the researchers from the application of the aforementioned algorithms on the analyzed data indicate that (C&RT) has minimal error in predicting machine failures, and thus was considered the optimal model.

H. Case study of: "Investigating Strategies and Parameters to Predict Maintenance of an Elevator System"

In this chapter we will focus on the article whose authors are: Jasmine Awatramani, Gaayan Verma, Nitasha Hasteer, and Rahul Sindhwani. This research paper is part of the Smart Innovation, Systems and Technologies book series [8]. In the paper, attention has been given to the elevator industry, as this industry, like many others, requires constant monitoring and regular maintenance to ensure adequate safety. In order to save costs and improve safety, the researchers decided to study the optimal maintenance policy for the elevator system. The researchers focused primarily on diagnostic and prognostic techniques, managing scheduled tasks with access to limited data, and predicting the RUL (Remaining Useful Life) of the machine.

The construction of the fault consists of a large number of different components, each of which has a certain impact on the proper operation of the machine. Such components include, for example, Ball-bearings present in the driving pulley of the elevator. The authors of the article decided to focus their attention on these, as well as the readings from vibration and humidity sensors. The first analysis of the database showed a steady decline in RPM. When they reach a certain critical value, this is a sign that the elevator is the most vulnerable to failure. In addition, from the analysis of vibration sensors, it is possible to determine the health of the elevator motor.

In the following part, the authors made the classification and for this purpose they decided to use the Random Forest method. In this technique, by using a large number of decision trees, the final category of the test object can be classified. Ultimately, each of the analyzed samples takes one of 3 classes: Good (0), Fair (1), and Poor (2). After classifying all the samples and assigning them to the corresponding classes, the researchers obtained a score of 91.5%, which shows the high accuracy of the classification. In the end, the researchers concluded that the ball bearing feature proved to be more significant in the analysis than the vibration feature, which may have been due to the frequent fluctuation of vibration readings in the studied dataset.

In conclusion, the authors of the article conducted an experiment to predict the maintenance of an elevator system. They demonstrate that PMx provides an opportunity not only to prevent future accidents, but also to save the lives of many people who could become accidental victims of a sudden machine failure.

I. Case study of "A Comparative Study of State-of-the-Art Machine Learning Algorithms for Predictive Maintenance"

The analyzed article was created by Luis Silvestrin, Mark Hoogendoorn and Ger Koole [9].

The dataset used to generate the analysis has been downloaded from UCI website and is available under the following link: UCI data set. The data applies to the condition assessment of a hydraulic test rig based on multi-sensor data, where 4 types of faults with several grades of severity that impede the selective quantification are distinguished.

In this project, the main task the researchers decided to tackle was to test the application of deep learning methods such as LSTM (Long Short-term Memory) and TCN (Temporal Convolutional Networks) in scenarios with scarce sensor data and then compare the results obtained with simpler machine learning models with feature engineering (like Random Forest and Decision Tree). The researchers decide to use Hydraulic System sensor dataset in order to evaluate the performance of different machine learning techniques in a real-life predictive maintenance scenario. There were 3 different Machine Learning techniques used in order to perform the desired analysis: Random Forest, Decision Tree and K-nearest neighbors combined with simple (time and frequency-based) features extracted from the sensors. The prediction target for the experiment was the internal pump leakage (what might be labeled as: no leakage, weak leakage, and severe leakage, giving the results in 3 different distinct classes). For the purpose of training the algorithms, the research group decided to divide the dataset into training and test data. The hyperparameters of the various algorithms were optimized based on experiments conducted on the training set.

Turning to the techniques used in the experiment, in the case of Decision Tree, the researchers decided to use the Gini impurity considering this as a form of criterion for selecting the allocation attribute for each node of the tree. For the Random Forest, the researchers built 10 decision trees with the same criterion as for the Decision Tree. In the technique called the k-nearest neighbor, the classifier was trained using k value equal to 1, making it the nearest neighbor classifier. All the algorithms used were compared based on the classification error obtained on the test set.

The results of the analysis are different for the traditional machine learning algorithms and the deep learning methods. The second group of methods which consist of TCN and LSTM present a higher classification error compared to the baseline machine learning algorithms. Eventually it turned out that compared to the LSMT in the performed tasks, the TCN has obtained a better performance with fewer parameters.

When it comes to the traditional machine learning algorithms, both the Random Forest and an ensemble of Decision Trees show lower error compared to a single tree which has similar performance as K-nearest neighbors. Moreover, these methods might learn with fewer examples than deep learning methods.

This provides a kind of confirmation that, compared to deep learning methods (LSTM and TCN) which do not use future engineering, traditional machine learning methods combined with simple feature engineering techniques (such as Random Forest or Decision Tree) can be a very good choice for analyses where the amount of data is limited.

J. Case study of "Prediction of Failures in the Air Pressure System of Scania Trucks using a Random Forest and Feature Engineering"

The next and final article discussed, is written by Christopher Gondek, Daniel Hafner, and Oliver R. Sampson [10] and concerns predicting air pressure system failures in Scania trucks. The authors decided to use Random Forest and Feature Engineering to perform an analysis whose results would enable them to minimize the cost of maintaining the air pressure system in these trucks.

The experiment carried out is part of the Industrial Challenge for IDA 2016, the intention of which was to be able to predict failures even before they occur in the Air Pressure System (APS) in Scania trucks which would simultaneously mean reducing maintenance costs. To perform this experiment, the researchers decided to use feature creation on histograms. Of the 171 columns in the database, one column is a class column. The database had 60,000 rows, of which 1,000 belonged to the positive class. A total of 7 histograms were created, each with 10 bins, whose analysis, according to the researchers, showed that each histogram indicated the age of the Air Pressure System. To visually inspect and analyze the database, the researchers used the following methods: Box plots to get an overview of the variance of the values; Correlation matrices for identifying features that correlate; Scatter plots to see how the classes are spread; and Radar charts to recognize outliers. For the experiment conducted on the histograms prepared for this, two distance functions were used: - 2-distance, which is a binary comparison of the observed value with the expected value. - Earth Mover's Distance - gives the opportunity to transform one histogram into another. Both of these functions were used by researchers to calculate distances to four different distributions, such as Mean distribution of the positive and negative examples, Normal distribution and Mirrored normal distribution. In order to negate the strong correlation of the above distances with the bin sum, the authors decided to normalize the histograms by their sum. Finally, the calculated features yielded the creation of 282 dimensions (without the class column, due to their potentially strong correlation). The result was feature selection: first the researchers ranked the features according to their expressiveness and then tested the performance of sets of features that differed in size. Once the features were properly ranked, the researchers moved on to calculating the cost of the predictive model using a varying number of dimensions. To do this, they trained Random Forest and made class predictions using 10-fold cross-validation and calculated average costs. This analysis allowed the researchers to make some conclusions, namely that not all dimensions are needed, which slowed down the reduction of a certain number of dimensions - which also automatically reduces the costs associated with the study. Training the Random Forest algorithms had an important purpose. The problem was that the cost of a false negative is 50 times higher than that of a false positive and Random Forest algorithms usually try to minimize the prediction error, assuming that all errors have an equally high cost. To solve the problem, the researchers decided to establish prediction confidence thresholds for each subset of features, which were further modified by 1 percent. After analyzing the results of the method used, the researchers found that for most cases the best threshold was 95 percent. After the analyses, the researchers came to the common conclusion that undoubtedly, conducting this type of data analysis in the field provides an opportunity to strengthen and significantly improve the effectiveness over regular inspections of each truck up to the point of failure. It is extremely important for any company to forecast failures, preventing them from occurring, as this definitely strengthens the level of safety and significantly reduces the costs associated with an unexpected failure.

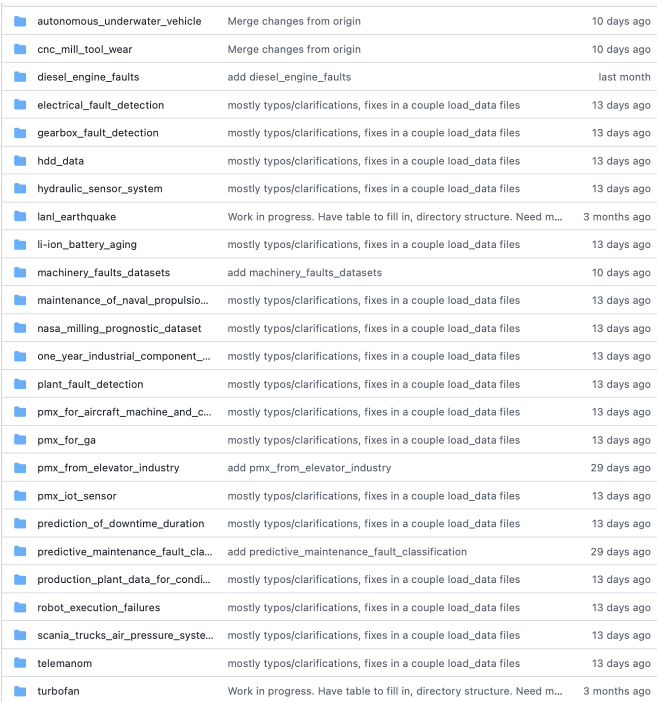
K. Summary of case studies

To sum up all the case studies of the chosen articles, undoubtedly every article contains different but also essential thoughts, analysis, and information related to predictive maintenance. The first discusses the latest trends and implementations in PMx. The next 4 articles are strictly about implementation of predictive models using datasets from the aviation. Due to the fact that we don't have access to the datasets used in some of these, we thought that it would be best to summarize the articles to provide some information about using aviation datasets to create PMx models. The next article is from the machine industry and uses a dataset taken from a vibration sensor in an exhaust fan to show how important well-developed early failure detection can be. The next article is about IoT-based predictive maintenance. The article and dataset in it is from the auxiliary automotive industry, attempting to create a model to enable the prediction of sudden failures in industrial machinery. The seventh article is related to the elevator industry and aims to predict the remaining useful life of the machine. The next article is related to hydraulic systems, using the sensor data of a hydraulic test rig in order to predict the internal pump leakage. It examines different machine learning techniques and deep learning methods and then compares the results. The last article is from the Scania trucks industry and analyzes the trucks' air pressure system, in order to predict the damage of the system and minimize the costs related to unforeseen repairs as much as possible.

V. GitHub repository description

For this project, one of the biggest tasks was to create a GitHub repository for a large collection of Predictive

Maintenance benchmark datasets found on the internet.



autonomous_underwater_vehicle	Merge changes from origin	10 days ago
cnc_mill_tool_wear	Merge changes from origin	10 days ago
diesel_engine_faults	add diesel_engine_faults	last month
electrical_fault_detection	mostly typos/clarifications, fixes in a couple load_data files	13 days ago
gearbox_fault_detection	mostly typos/clarifications, fixes in a couple load_data files	13 days ago
hdd_data	mostly typos/clarifications, fixes in a couple load_data files	13 days ago
hydraulic_sensor_system	mostly typos/clarifications, fixes in a couple load_data files	13 days ago
lanl_earthquake	Work in progress. Have table to fill in, directory structure. Need m...	3 months ago
li-ion_battery_aging	mostly typos/clarifications, fixes in a couple load_data files	13 days ago
machinery_faults_datasets	add machinery_faults_datasets	10 days ago
maintenance_of_naval_propulsio...	mostly typos/clarifications, fixes in a couple load_data files	13 days ago
nasa_milling_prognostic_dataset	mostly typos/clarifications, fixes in a couple load_data files	13 days ago
one_year_industrial_component_...	mostly typos/clarifications, fixes in a couple load_data files	13 days ago
plant_fault_detection	mostly typos/clarifications, fixes in a couple load_data files	13 days ago
pmx_for_aircraft_machine_and_c...	mostly typos/clarifications, fixes in a couple load_data files	13 days ago
pmx_for_ga	mostly typos/clarifications, fixes in a couple load_data files	13 days ago
pmx_from_elevator_industry	add pmx_from_elevator_industry	29 days ago
pmx_iot_sensor	mostly typos/clarifications, fixes in a couple load_data files	13 days ago
prediction_of_downtime_duration	mostly typos/clarifications, fixes in a couple load_data files	13 days ago
predictive_maintenance_fault_cla...	add predictive_maintenance_fault_classification	29 days ago
production_plant_data_for_condi...	mostly typos/clarifications, fixes in a couple load_data files	13 days ago
robot_execution_failures	mostly typos/clarifications, fixes in a couple load_data files	13 days ago
scania_trucks_air_pressure_syste...	mostly typos/clarifications, fixes in a couple load_data files	13 days ago
telemanom	mostly typos/clarifications, fixes in a couple load_data files	13 days ago
turbofan	Work in progress. Have table to fill in, directory structure. Need m...	3 months ago

Fig. 1. GitHub repository view

At the moment, the repository contains 26 databases on various topics. Below we present a short description of all datasets included in repository:

1) Prediction of downtime duration of a factory in the field of Predictive Maintenance [11] – this dataset might be used to predict the downtime duration of various factories, which can result in identifying the factories that are most prone to downtime.
2) CNC Mill Tool Wear – this dataset can be used in the following experiments [12]:
   - Tool wear detection: you can use supervised binary classification to identify worn and unworn cutting tools. 18 experiments were conducted with an unworn tool, whereas 10 were run with a worn tool.
   - Detection of inadequate clamping: in this case, the data would be used in order to detect the condition when a workpiece is not being held in the vise with sufficient pressure to pass visual inspection. The experiments were run with the pressure values of 2.5, 3 and 4 bar. Moreover this data could also be used to detect the conditions at which a critical point occurs which would prevent the machining operation from completing.
3) Diesel Engine Faults [13]– the dataset could be used for fault diagnosis in Diesel engines, through the analysis of the variation of the pressure curves inside the cylinders and the torsional vibration response of the crankshaft.
4) Electrical Fault Detection and Classification [14] – this dataset consists of a collection of line currents and voltages for different fault conditions. The faults on electrical power system transmission lines are supposed to be first detected and then be classified correctly and should be cleared as fast as possible. The protection system used for a transmission line can also be used to initiate the other relays to protect the power system from outages. A good fault detection system provides an effective, reliable, fast and secure relaying operation.
5) Fighter Aircraft Flight Logs [15]– this dataset consists of data collected from flight logs from 3 aircraft from the same fleet. It can be used to create a predictive maintenance model in order to predict possible failures that may happen to the aircraft components based on the sensor measurements and their comparison to the threshold values.
6) Gas Emissions from Gas Turbines [16]– the dataset contained in this folder might be used to predict the possible gas emissions (NO, COx) from a gas turbine. This could be further used to create a predictive model that would predict when emission values would exceed the permissible standards, which would provide an opportunity to plan maintenance in advance to avoid exceeding the permitted emission threshold.
7) Gearbox Fault Diagnosis [17] - This dataset might be used to cover the basics for Predictive Maintenance in industrial facilities in order to effectively predict the potential failure of the gearbox. This would enable maintenance to be planned well in advance, avoiding unplanned downtime of the machine.
8) HDD Data [18] – This data comes from different hard drives. It can be used in order to predict the potential failure of the hard drives.
9) Hydraulic System Sensor [19] - This data is taken from the sensors of hydraulic test equipment to evaluate its technical condition. Can be used to perform condition monitoring of a hydraulic rig.
10) Li-ion Battery Aging [20] – This dataset might be used in order to develop prognostic algorithms. The aim is to be able to manage the uncertainty (caused by differentials between the same state-of-life for 2 cells at the same cycle index) which is representative of actual usage, and make reliable predictions of Remaining Useful Life in both the End-of-Discharge (EOD) and End-of-Life (EOL) contexts.
11) Machine PMx Classification [21] - This dataset contains synthetic data that has been created for real predictive maintenance purposes in the industry field. The purpose of this dataset is to predict machine failure and type.
12) Machinery Faults Datasets [22] – This dataset is composed of 1951 multivariate time-series acquired by sensors on a SpectraQuest's Machinery Fault Simulator (MFS) Alignment-Balance-Vibration (ABVT) and enables prediction of induction motor faults.

13) Maintenance of Naval Propulsion Plants [23] - The dataset consists of records taken from a sophisticated simulator of Gas Turbines and can be used to investigate the problem of performing Conition-Based Maintenance through the use of Data-Driven Models.
14) One Year Industrial Component Degradation [24] - The purpose of this dataset is to present the component degradation process over the course of a year. Based on such analysis you would be able to check if the component has been replaced at some point, you might check if the wear can be accurately predicted, and you might make a prediction of the RUL (Remaining Useful Life) in order to determinate the maintenance windows.
15) PMx IoT Sensor [21] - The database is derived from a proxy of automotive components, such as motors, rotors and heat exchangers, whose failure brings the entire assembly line to a halt. It is therefore important to predict the failure of these components in order to avoid such situations in the future and thus improve product quality and save energy spent on machine work.
16) PMx for GA [2] – This is dataset for predictive maintenance for General Aviation. The main problem that might be taken into consideration while using this dataset is classifying whether the flights in question fell into the problematic (i.e., those that resulted in forced and unscheduled maintenance) or non-problematic (i.e., post maintenance) categories. The dataset can be used in order to create models that could detect the features that are relevant to classification.
17) Plant Fault Detection [25] - The dataset is about fault detection and prognostics in industrial plant monitoring. The aim of this dataset is to provide the ability to create a model to detect plant faults.
18) Predictive Maintenance Fault Classification – This dataset can be used to train different types of models for fault diagnosis in supervised learning (such as SVM, random forest, k-nearest neighbour and H2O's AutoML model).
19) Predictive Maintenance for Aircraft Machine and Components [26] – The aim of this dataset is to predict failures due to certain components of a machine in a 24 hour period.
20) Predictive Maintenance from the Elevator industry [27] – This dataset comes from the elevator industry and can be useful in performing analysis of elevator doors that may raise the amount of equipment life cycles and reduce unplanned stops.
21) Production Plant Data for Condition Monitoring [28]- The dataset has been created in order to help predict the condition of an important component within production lines. This condition is essential for the function of the plant and the resulting product quality.
22) Robot Execution Failures [29]- This dataset was collected, defined and evaluated in order to improve classification accuracy. It consists of force and torque measurements on a robot after failure detection.
23) Scania Trucks Air Pressure System Failure Prediction [30] – This dataset can be used to create a predictive maintenance model to investigate the condition of trucks' air pressure system.
24) Solar Power Generation [31]– This dataset might be used to enable prediction of the power generation for next couple of days, identifying the need for panel cleaning/maintenance and identifying faulty or sub-optimally performing equipment.
25) Turbofan engine degradation system [32]– This is a dataset that can be used in order to predict the remaining useful life of a turbofan aircraft engine.
26) Pump sensor data for PM [33] - The aim of this dataset is to help detect any possible anomalies in pump behaviour, in order to stop the pump before it breaks down and more effectively manage critical components

Each of the corresponding folders in the repository contains a README file and 2 subfolders: dataset and documentation.

- The dataset subfolder contains data needed for research and analysis purposes.
- The documentation subfolder consists of the sources file containing links, citations and references to the original website from which the dataset has been downloaded, to the different articles relating to the dataset, and code that might be useful for working with this dataset.
- The README file contains a description of the dataset.

The repository is available here. Everyone in need of predictive maintenance data can use the repository without any restrictions.

VI. Methods to Search for Predictive Maintenance Datasets and Articles

To find the datasets, we searched a number of different websites with databases from various fields to find those that are applicable to predictive maintenance. Anyone who would like to expand their knowledge in the field of predictive maintenance, or would like to try their hand at finding other databases, may find the following websites very useful:

1) Kaggle - databases and discussions, code
2) UCI Machine Learning Repository - databases
3) Google scholar browser - science documents
4) Researchgate - science documents
5) IEEE Data Port - databases and articles
6) GitHub - databases, code
7) Emerald - science documents
8) ScienceDirect – science documents
9) Medium - articles, often with code analysis
10) Datasearch - databases
11) NASA Open data portal - databases
12) Springer Link - scientific documents
13) ProQuest - library with research documents
14) Towards Data Science – science documents

Each of these sites contains a wealth of valuable resources that can prove very useful in your search for information in various scientific fields, but especially in the field of predictive maintenance.

Our search for materials and databases made use of the above sites. UCI, Kaggle, GitHub and DataSearch are extremely helpful in sourcing various databases, many of which also come with a brief description or related articles. Kaggle and GitHub very often additionally contain code written as part of the analysis performed on the database in question. The other sites mentioned above provide an invaluable source of all kinds of existing articles in the field. After typing the phrase "predictive maintenance" (with other added keywords to tighten the scope if need be), we get a wide range of articles of interest. This opens the door to many interesting articles that are related to the chosen topic.

VII. Conclusions

Taking into account all of the articles examined here, it can be deduced that access to necessary data, the method of machine learning used, and the method of organizing and selecting data varies depending on the type of project, field and component/device. The methods which give the best results for any one experiment are very diverse and it is impossible to unambiguously determine which method is universal and works best, because each database, each component, sensor, and device is undeniably characterized by uniqueness. This makes each experiment require different choices of methods which give the most effective and accurate results of the analysis carried out.

References

[1] Şule Selçuk, "Predictive maintenance, its implementation and latest trends," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 231, pp. 1670 – 1679, 2017.

[2] H. Yang, A. LaBella, and T. Desell, "Predictive maintenance for general aviation using convolutional transformers," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 11, 2022, pp. 12 636–12 642.

[3] E. Burnaev, "On construction of early warning systems for predictive maintenance in aerospace industry," *Journal of communications technology and electronics*, vol. 64, no. 12, pp. 1473–1484, 2019.

[4] W. J. Verhagen and L. W. De Boer, "Predictive maintenance for aircraft components using proportional hazard models," *Journal of Industrial Information Integration*, vol. 12, pp. 23–30, 2018.

[5] P. Korvesis, S. Besseau, and M. Vazirgiannis, "Predictive maintenance in aviation: Failure prediction from post-flight reports," in *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, 2018, pp. 1414–1422.

[6] N. Amruthnath and T. Gupta, "A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance," in *2018 5th International Conference on Industrial Engineering and Applications (ICIEA)*, 2018, pp. 355–361.

[7] S. Nangia, S. Makkar, and R. Hassan, "Iot based predictive maintenance in manufacturing sector," *SSRN Electronic Journal*, 01 2020.

[8] J. Awatramani, G. Verma, N. Hasteer, and R. Sindhwani, *Investigating Strategies and Parameters to Predict Maintenance of an Elevator System*, 01 2022, pp. 323–332.

[9] L. P. Silvestrin, M. Hoogendoorn, and G. Koole, "A comparative study of state-of-the-art machine learning algorithms for predictive maintenance." in *SSCI*, 2019, pp. 760–767.

[10] C. Gondek, D. Hafner, and O. R. Sampson, "Prediction of failures in the air pressure system of scania trucks using a random forest and feature engineering," in *International Symposium on Intelligent Data Analysis*. Springer, 2016, pp. 398–402.

[11] A. Maity. (2021) Prediction of downtime duration of a factory.

[12] U. O. M. S. LAB. (2018) CNC Mill Tool Wear.

[13] D. Pestana. (2020) Diesel Engine Faults Features Dataset.

[14] M. Jamil, S. Sharma, and R. Singh, "Fault detection and classification in electrical power transmission system using artificial neural network," *SpringerPlus*, vol. 4, p. 334, 07 2015.

[15] (2022) AutonLab PMx Repository - Fighter Aircraft Dataset.

[16] U. M. L. Repository. (2019) Gas Emissions from Gas Turbines.

[17] Y. Pandya. (2018, 06) Gearbox fault diagnosis data. https://data.openei.org/submissions/623.

[18] Backblaze. (2013) Backblaze HDD Data Center.

[19] U. M. L. Repository. (2018) Condition monitoring of hydraulic systems Data Set .

[20] VaibhavBhujade. (2020) Li-ion Battery Aging.

[21] U. M. L. Repository. (2020) AI4I 2020 Predictive Maintenance Dataset .

[22] Mafaulda. (2022) Machinery Fault Database.

[23] U. M. L. Repository. (2014) Condition Based Maintenance of Naval Propulsion Plants Data Set .

[24] A. von Birgelen, D. Buratti, J. Mager, and O. Niggemann, "Self-organizing maps for anomaly localization and predictive maintenance in cyber-physical production systems," in *51st CIRP Conference on Manufacturing Systems (CIRP CMS 2018)*. CIRP-CMS, May 2018.

[25] T. Prognostics and H. M. Society. (2009-2022) Plant Fault Detection - PHM society .

[26] (2020) Microsoft Azure Predictive Maintenance .

[27] J. Awatramani, G. Verma, N. Hasteer, and R. Sindhwani, "Investigating strategies and parameters to predict maintenance of an elevator system," in *Computational and Experimental Methods in Mechanical Engineering*, V. V. Rao, A. Kumaraswamy, S. Kalra, and A. Saxena, Eds. Singapore: Springer Singapore, 2022, pp. 323–332.

[28] A. Von Birgelen, D. Buratti, J. Mager, and O. Niggemann, "Self-organizing maps for anomaly localization and predictive maintenance in cyber-physical production systems," *Procedia cirp*, vol. 72, pp. 480–485, 2018.

[29] U. M. L. Repository. (1999) .

[30] K. T. Selvi, N. Praveena, K. Pratheksha, S. Ragunanthan, and R. Thamilselvan, "Air pressure system failure prediction and classification in scania trucks using machine learning," in *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*. IEEE, 2022, pp. 220–227.

[31] (2020) https://www.kaggle.com/datasets/anikannal/solar-power-generation-data.

[32] A. Listou Ellefsen, E. Bjørlykhaug, V. Æsøy, S. Ushakov, and H. Zhang, "Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture," *Reliability Engineering System Safety*, vol. 183, pp. 240–251, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0951832018307506

[33] (2018) https://www.kaggle.com/datasets/nphantawee/pump-sensor-data.

Survival Analysis on Chest Radiographs with Deep Learning

Mingzhu Liu[1], Chirag Nagpal[2], Artur Dubrawski[3]

***Abstract*— Chest radiography is a common diagnostic imaging test. An automated detector would improve workflow prioritization and clinical decision, thus supporting large-scale screening. Most of the existing methods focus on the binary outcomes for patients, i.e. all-cause mortality. In this work, we implement and compare different models that utilize time-to-event and censor indicators for survival analysis. The models consist of a convolutional neural network model and a survival model. The architectures take a chest radiograph as input and predict long-term mortality.**

***Index Terms*— Computer Vision**

I. INTRODUCTION

Chest radiography is a common diagnostic imaging test. The demand for it worldwide is growing rapidly [1]. It is used for identifying various diseases, such as pneumonia, pneumothorax, and heart failure. However, chest radiographs could be hard even for physicians to interpret without the assistance of radiologists [2]. Therefore, a model that predicts mortality of patients could help with clinical decision making.

Despite the importance of chest radiography, there has been limited research on predicting survival with it. Deep learning involves the use of neural networks to learn patterns existing in data. There has been extensive research on detecting abnormalities in chest radiographs using deep learning, but little research on applying existing survival models to chest radiographs has been done.

Survival analysis involves estimating the time T until an event E occurs. When an event is observed, E is equal to 1, and T is the time-to-event. When it is censored, E is equal to 0, and T is the time to the last contact with the patient. In this paper, we study the probability that a patient survives some time t, given covariates or explanatory variables Z. This is expressed as $P(T > t|Z)$. The risk of a patient not surviving time t is $1 - P(T > t|Z)$.

In this work, we propose multiple models that use a convolutional neural network (CNN) and a survival model to predict survival probabilities using chest radiographs. We also compare their performance using various metrics. We expect that the CNN learns a lower-dimensional representation of chest radiographs, and the survival model learns the survival probability based on the representation.

[1]Mingzhu Liu is with Computer Science and Engineering, University of Michigan, 2260 Hayward Street, USA mingzliu@umich.edu

[2]Chirag Nagpal is with the Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA chiragn@andrew.cmu.edu

[3]Artur Dubrawski is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA awd@andrew.cmu.edu

II. RELATED WORKS

A. Deep Learning with Chest Radiographs

Efforts have been made for learning lower-dimensional representations of chest radiographs using CNNs. CheXNet [3] is a DenseNet121 model [4] that predicts probabilities of 14 diseases present in a chest radiograph. Similarly, CheXaid [5] is a DenseNet121 model that uses chest radiographs for diagnosis of tuberculosis for patients infected with HIV. Besides the probability of tuberculosis present in the image, the model outputs six additional covariates, such as age and white blood cell count. A DenseNet model is used in [6] as an encoder that takes a chest X-ray image as input and learns a representation of the input. Long short-term memory is used as a decoder that takes the representation as input and learns the probabilities of 14 pathologies.

CXR-risk CNN [7] was the first to use deep learning on chest radiographs to predict patient risk. It uses an Inception-v4 architecture to predict risk score (very low, low, moderate, high, and very high). The risk scores indicate the risk of long-term all-cause mortality.

B. Survival Analysis

Multiple models for survival analysis exist, such as Cox Proportional Hazards [8], Deep Survival Machines [9], and Deep Cox Mixtures [10].

1) Cox Proportional Hazards: Cox Proportional Hazards (CPH) makes the proportional hazards assumption that the ratio between two hazards is constant over time [8]. The hazard $\lambda(t)$ as a function of some time t is defined as the probability that given a patient survives t, the patient will not survive $t+\delta$, and δ approaches 0 (Equation 1).

$$\lambda(t) = \lim_{\delta \to 0} \frac{P(t+\delta > T \geq t | T \geq t)}{\delta} \tag{1}$$

In DeepSurv [11], the risk function $\hat{h}_\theta(x)$ of an individual's covariates x is estimated by multi-layer perceptron, where θ is its weights. The loss function is the negative log partial likelihood, Equation 2, where $\mathscr{R}(T_i)$ is the set of patients who have not experienced the event at time T_i.

$$L_{CPH}(\theta) = -\sum_{i:E_i=1} \left(\hat{h}_\theta(x_i) - \log \sum_{j \in \mathscr{R}(T_i)} e^{\hat{h}_\theta(x_j)} \right) \tag{2}$$

2) Deep Survival Machines: Deep Survival Machines (DSM) [9] is a parametric neural network model, without the assumption of proportional hazards. The distribution of survival times conditioned on input covariates is modeled as a mixture of k distributions. The distributions are required to have closed-form cumulative distribution function solutions.

The representation of input features, such as radiologists' impressions and demographic information, are acquired by using multilayer perceptron. From the representation, weights and parameters of the distributions are learned. The distribution of survival times are estimated from the k distributions.

III. Methodology

Our models contain an encoder and a survival models. In Subsection *A*, we introduce the encoder. In Subsection *B*, we introduce survival models. In the rest of this section, we introduce our models.

A. Encoder

We use a CNN model from the TorchXRayVision Python package [12]. It contains DenseNet121 models trained on various chest X-ray datasets, such as CheXpert and MIMIC-CXR. The inputs to the models are grayscale chest X-ray images and outputs from the models are prbabilities of 18 pathologies present in the input image, which we refer to as radiologists impressions or concepts.

We use the `densenet121-res224-all` model in the package. All outputs of this model are trained, whereas for other models, not all outputs are meaningful because some targets do not exist in the dataset. We finetuned the trained model on our dataset.

Binary cross entropy loss, Equation 3, is used. The weight w_n for a concept n is calculated by the inverse of the proportion of positive labels.

$$L_{encoder} = -\sum_n w_n \left(y_n \log \sigma(x_n) + (1-y_n)\log(1-\sigma(x_n))\right) \tag{3}$$

B. Survival Models

Survival models take concepts and demographic data as inputs and output survival probabilities. We experiment with DSM, CPH, and a binary survival model. We use the binary survival model as a baseline. The models are from the auton-survival package [13].

Given a time horizon, the binary survival model takes concepts and demographic data as inputs and outputs one score. We ignored all data points with a time-to-event less than the specified time horizon and a censor indicator equal to 0. All patients survive the given time horizon belong to the positive class. The loss function is the binary cross entropy loss in Equation 3, where $w_n = 1$. During inference time, we obtain the predictions of the model on the training dataset. We split the predictions into a certain number of bins. For data in each bin, we fit a Kaplan–Meier estimator using time-to-event and censor indicator. For each data point in the test dataset, the survival probability is estimated by the prediction of the corresponding Kaplan–Meier estimator.

C. Ground-truth concepts + survival models

Model architecture is illustrated in Fig. 1. We use ground-truth (GT) radiologists' impressions and demographic data as inputs to the survival models.

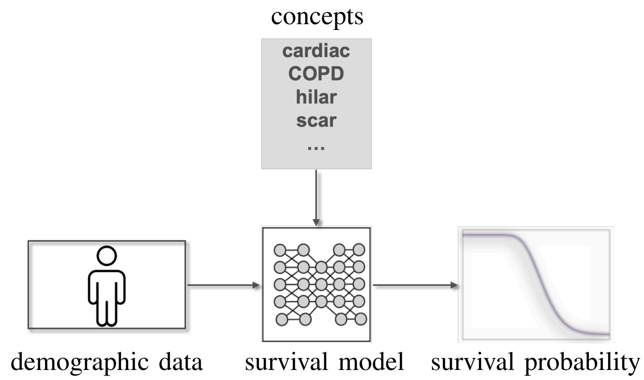


Fig. 1: Ground-truth concepts + survival model

D. Independent Models

Model architecture is illustrated in Fig. 2. We use chest radiographs as inputs to the encoder, and trained the encoder using ground-truth radiologists' impressions. We use the trained encoder, and obtained predicted concepts on chest radiographs. We use the predictions and demographic data to train survival models.

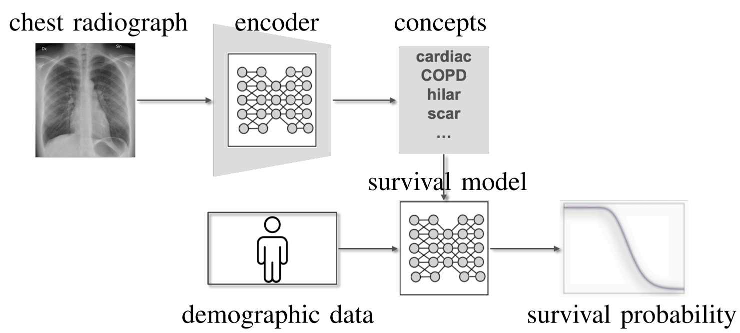


Fig. 2: Independent model

E. Jointly trained Models

Model architecture is illustrated in Fig. 3. We build one model that consists of an encoder, a multi-layer percecptron, and a survival model. A chest radiograph is the input to the encoder and the output is a vector that represent the concepts. We concatenate the concepts and demographic data, and pass them to the multi-layer percecptron. The output of the multi-layer percecptron is the input to the survival model. The loss function is Equation 4, a sum of concept loss and survival loss, where concept loss is a binary cross entropy loss (Equation 5) that compares predicted concepts and ground-truth radiologists' impressions, and survival loss is the loss of the survival model.

$$L_{joint} = \alpha L_{concepts} + L_{survival} \tag{4}$$

$$L_{concepts} = -\sum_n w_n \left(y_n \log \sigma(x_n) + (1-y_n)\log(1-\sigma(x_n))\right) \tag{5}$$

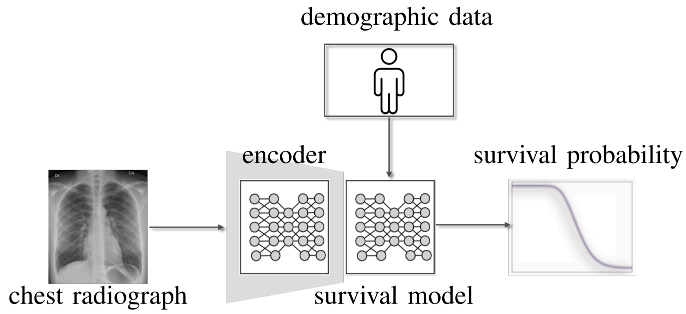


Fig. 3: Jointly trained models

1) End-to-End models: End-to-end models are jointly trained models when $\alpha = 0$. This means we do not train the models based on ground truth radiologists' impressions. Instead, we only train the models based on time-to-event and censor indicators.

2) Concept bottleneck models: Inspired by [14], we build concept bottleneck models by setting $\alpha \neq 0$. This means we train the models based on both ground truth radiologists' impressions and time-to-event and censor indicators.

IV. Experiments

A. Dataset

Data from the Prostate, Lung, Colorectal, and Ovarian Cancer Screening Trial (PLCO) [15] is used in this study. In this study, we use chest radiographs, radiologist's impressions, demographic data, time-to-death, and censor indicator. Demographic data includes data such as sex, race, and age. Radiologist's impressions indicates whether abnormalities exist, such as pleural fluid and COPD.

Each image is in grayscale and preprocessed as follows. Each image is resized to be 256 by 256 pixels. For images used for testing the model, they are cropped into four corners and the central crop. Predictions are the mean of the predictions of each crop. For images used for training the model, they are cropped at a random position. The resulting images are 224 by 224 pixels.

There are 89643 data points in total, and they are split into 60% training, 20% validation, and 20% test.

B. Evaluation Metrics

We evaluate the performance of models using brier score, concordance index, area under the receiver operating characteristic curve (AUC), and expected calibration error (ECE) on 2-year, 5-year, and 10-year time horizons.

1) Brier score: Brier score measures the average squared distance between the actual survival probability and predicted survival probability at a given time horizon. Brier score is given in Equation 6, where N is the number of data points, f_t is the predicted probability of the outcome, o_t is the actual outcome. Larger Brier score indicates greater error.

$$BS = \frac{1}{N}\sum_{t=1}^{N}(f_t - o_t)^2 \tag{6}$$

2) Concordance index based on inverse probability of censoring weights: Concordance index measures how well the predicted probabilities are ranked. Each pair of data is compared in terms of time-to-event and risk probability.

For each pair of patients i and j, their predicted risk scores are η_i and η_j respectively, and their ground-truth time-to-event are T_i and T_j. We consider the following situations for each pair.

If they are both censored, we ignore this pair.

If they are both not censored, they are a concordant pair if $\eta_i > \eta_j$ and $T_i < T_j$, and they are a discordant pair if $\eta_i > \eta_j$ and $T_i > T_j$.

If T_i is censored but T_j is not, we consider the following situations. If $T_i < T_j$, we ignore this pair. If $T_i > T_j$, they are a concordant pair if $\eta_i < \eta_j$, and they are a discordant pair if $\eta_i > \eta_j$.

Concordance index is given in Equation 7, where n_c is the number of concordant pairs, and n_d is the number of discordant pairs.

$$C = \frac{n_c}{n_c + n_d} \tag{7}$$

Concordance index based on inverse probability of censoring weights [16] uses Kaplan-Meier estimator to estimate the censoring distribution, and weights are applied to each pair based on the inverse of it.

3) AUC: Receiver operating characteristic (ROC) curve is plotted as the true positive rate v.s. the false positive rate (recall) when different thresholds are applied. AUC is the area under the curve. A greater AUC indicates that predictions are well-ranked, meaning that the model gives positive samples higher scores than negative samples.

4) ECE: Calibration measures how well predicted probabilities match true probabilities. Larger ECE indicates larger miscalibration. ECE is presented in Equation 8. Dataset is split into B bins, and for data in each bin (X_b), a prediction is made by a Kaplan Meier estimator (KM) fitted using data in the bin. An error is calculated by the absolute value of the difference between the prediction and the mean of the risk scores (r_b). ECE is calculated by the weighted sum of the error.

$$ECE = \sum_{b=1}^{B}\frac{n_b}{N}|KM(X_b) - mean(r_b)| \tag{8}$$

C. Implementation Details

All experiments were run on NVIDIA RTX A6000, with Python 3.9.12 and Pytorch 1.11.0.

We performed hyperparameter search. The best independent models were the ones with lowest validation survival loss. The best jointly trained models were the ones with lowest validation brier score. For CPH and binary models, we searched for layer sizes in $[[], [64], [64, 64], [128], [128, 128]]$. For DSM models, besides layer sizes, we searched for the number of underlying distributions ks in $[2, 3, 4, 6]$ and temperatures in $[1, 100, 500, 1000]$. For concept bottleneck models, we searched for αs in $[10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}]$. The best configurations are presented in Table I.

Adam optimizer with a learning rate of $3*10^{-4}$ was used for all models. The distribution for all DSM models was Weibull distribution. The batch size for independent survival models was 128. The training batch size for jointly trained models and independent encoders was 128, and validation batch size was 64. Jointly trained models and independent encoders were trained for 10 epochs, with patience equal to 3.

		layer size	k	temperature	α
GT + survival	DSM	[64]	3	500	N/A
	CPH	[]	N/A	N/A	
	binary	[]			
predicted concepts + survival	DSM	[64]	4	1	N/A
	CPH	[64]	N/A	N/A	
	binary	[]			
end-to-end	DSM	[64]	4	1	N/A
	CPH	[64]	N/A	N/A	
	binary	[]			
concept bottleneck	DSM	[64]	4	1	10^{-1}
	CPH	[64]	N/A	N/A	10^{-1}
	binary	[]			10^{-3}

TABLE I: Hyperparameter configurations

D. Results

Test performance of ground-truth concepts + survival models are presented in Table II. Test performance independent models are presented in Table III. Test performance of end-to-end models are presented in Table IV. Test performance of concept bottleneck models are presented in Table V.

	Brier Score			**Concordance Index**		
	2-year	5-year	10-year	2-year	5-year	10-year
DSM	0.0141	0.0421	0.1036	0.7541	0.7466	0.7356
CPH	0.0141	0.0423	0.1042	0.7512	0.7427	0.7318
Binary	0.0141	0.0424	0.1039	0.7514	0.7456	0.7343

	AUC			**ECE**		
	2-year	5-year	10-year	2-year	5-year	10-year
DSM	0.7561	0.7522	0.7516	0.0056	0.0081	0.0118
CPH	0.7533	0.7483	0.7474	0.0034	0.0076	0.0130
Binary	0.7534	0.7514	0.7501	0.0042	0.0126	0.0199

TABLE II: Test performance of ground-truth concepts + survival models

We compare the performance of ground-truth concepts + survival models and independent models in Fig. 4. Independent models are better than models using ground truth concepts with respect to all metrics except ECE of binary and CPH models on 2-year horizon and CPH models on 5-year horizon. In general, binary models perform worse than their DSM and CPH counterparts.

We compare the performance of jointly trained models and independent models in Fig. 5. In general, binary models perform worse than their DSM and CPH counterparts. Joint DSM and CPH models perform better than their independent counterparts in terms of AUC and concordance index, suggesting that the models are good at ranking the samples.

	Brier Score			**Concordance Index**		
	2-year	5-year	10-year	2-year	5-year	10-year
DSM	0.0140	0.0411	0.1001	0.7719	0.7686	0.7494
CPH	0.0140	0.0413	0.1007	0.7727	0.7660	0.7480
Binary	0.0140	0.0416	0.1009	0.7574	0.7625	0.7463

	AUC			**ECE**		
	2-year	5-year	10-year	2-year	5-year	10-year
DSM	0.7742	0.7749	0.7659	0.0054	0.0062	0.0083
CPH	0.7749	0.7722	0.7644	0.0048	0.0088	0.0120
Binary	0.7593	0.7687	0.7627	0.0046	0.0107	0.0145

TABLE III: Test performance of independent models

	Brier Score			**Concordance Index**		
	2-year	5-year	10-year	2-year	5-year	10-year
DSM	0.0141	0.0412	0.1006	0.7795	0.7711	0.7534
CPH	0.0140	0.0411	0.1001	0.7775	0.7713	0.7543
Binary	0.0141	0.0425	0.1060	0.7256	0.7243	0.7071

	AUC			**ECE**		
	2-year	5-year	10-year	2-year	5-year	10-year
DSM	0.7817	0.7775	0.7704	0.0056	0.0078	0.0173
CPH	0.7797	0.7776	0.7711	0.0028	0.0067	0.01639
Binary	0.7272	0.7293	0.7204	0.0066	0.0180	0.0483

TABLE IV: Test performance of end-to-end models

However, in terms of ECE, joint DSM and CPH models perform worse, suggesting that the predictions are not well calibrated.

End-to-end models and concept bottleneck models achieve comparable performance. As [14] suggests, concept bottleneck models are powerful in that humans are able to intervene on the model. For instance, if an radiologist does not agree with the predicted survival probability, he/she could examine predicted concepts, change those that he/she thinks are wrong, and see what survival probability the model gives.

E. Future Work

Our model can be used to perform counterfactual phenotyping, where groups of individuals are identified that belong to underlying clusters and demonstrate heterogeneous treatment effects [17]. This may provide insights into measuring the effects of an intervention.

V. CONCLUSION

In this paper, we propose models that take chest radiographs as inputs and output survival probabilities. We compare their performance in terms of Brier score, concordance index, AUC, and ECE.

For further work, we will apply bootstrapping and obtain confidence intervals of model performances.

ACKNOWLEDGMENT

I would like to thank Rachel Burcin, John M. Dolan, and other members of the Robotics Institute Summer Scholars

	Brier Score			Concordance Index			AUC			ECE		
	2-year	5-year	10-year	2-year	5-year	10-year	2-year	5-year	10-year	2-year	5-year	10-year
DSM	0.0141	0.0411	0.0100	0.7780	0.7701	0.7567	0.7802	0.7765	0.7567	0.0069	0.0080	0.0124
CPH	0.0140	0.0409	0.0991	0.7772	0.7697	0.7557	0.7794	0.7761	0.7726	0.0031	0.0056	0.01489
Binary	0.0141	0.0425	0.1066	0.7164	0.7168	0.7007	0.7180	0.7217	0.7137	0.0090	0.0178	0.0494

TABLE V: Test performance of concept bottleneck models

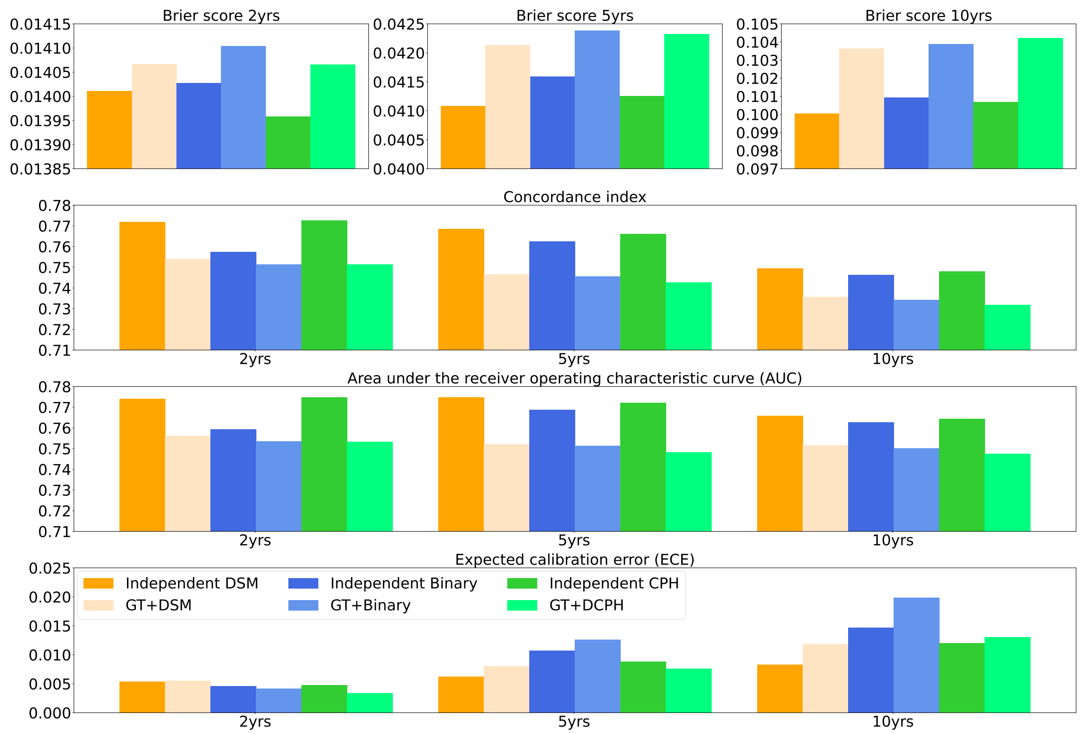


Fig. 4: Independent models and ground-truth + survival models

(RISS) program at Carnegie Mellon University for their support. Special thanks to my mentors Dr. Artur Dubrawski and Chirag Nagpal and the Auton Lab.

References

[1] K. T. Evans, "Radiology now. the radiologist's dilemma," *The British Journal of Radiology*, vol. 50, no. 593, pp. 299–301, 1977, pMID: 861448. [Online]. Available: https://doi.org/10.1259/0007-1285-50-593-299

[2] Z. Al aseri, "Accuracy of chest radiograph interpretation by emergency physicians," *Emergency Radiology*, vol. 16, no. 2, p. 111, 2008. [Online]. Available: https://doi.org/10.1007/s10140-008-0763-9

[3] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Y. Ding, A. Bagul, C. P. Langlotz, K. S. Shpanskaya, M. P. Lungren, and A. Y. Ng, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *CoRR*, vol. abs/1711.05225, 2017. [Online]. Available: http://arxiv.org/abs/1711.05225

[4] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016. [Online]. Available: http://arxiv.org/abs/1608.06993

[5] P. Rajpurkar, C. O'Connell, A. Schechter, N. Asnani, J. Li, A. Kiani, R. L. Ball, M. Mendelson, G. Maartens, D. J. van Hoving, R. Griesel, A. Y. Ng, T. H. Boyles, and M. P. Lungren, "Chexaid: deep learning assistance for physician diagnosis of tuberculosis using chest x-rays in patients with hiv," *npj Digital Medicine*, vol. 3, no. 1, p. 115, 2020. [Online]. Available: https://doi.org/10.1038/s41746-020-00322-2

[6] L. Yao, E. Poblenz, D. Dagunts, B. Covington, D. Bernard, and K. Lyman, "Learning to diagnose from scratch by exploiting dependencies among labels," *CoRR*, vol. abs/1710.10501, 2017. [Online]. Available: http://arxiv.org/abs/1710.10501

[7] M. T. Lu, A. Ivanov, T. Mayrhofer, A. Hosny, H. J. W. L. Aerts, and U. Hoffmann, "Deep Learning to Assess Long-term Mortality From Chest Radiographs," *JAMA Network Open*, vol. 2, no. 7, pp. e197416–e197416, 07 2019. [Online]. Available: https://doi.org/10.1001/jamanetworkopen.2019.7416

[8] D. R. Cox, "Regression models and life-tables," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 34, no. 2, pp. 187–220, 1972. [Online]. Available: http://www.jstor.org/stable/2985181

[9] C. Nagpal, X. Li, and A. Dubrawski, "Deep survival machines: Fully parametric survival regression and representation learning for censored data with competing risks," *CoRR*, vol. abs/2003.01176, 2020. [Online]. Available: https://arxiv.org/abs/2003.01176

[10] C. Nagpal, S. Yadlowsky, N. Rostamzadeh, and K. Heller, "Deep cox mixtures for survival regression," in *Machine Learning for Healthcare Conference*. PMLR, 2021, pp. 674–708.

[11] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger, "DeepSurv: personalized treatment recommender system using a cox proportional hazards deep neural network," *BMC Medical Research Methodology*, vol. 18, no. 1, feb 2018. [Online]. Available: https://doi.org/10.1186%2Fs12874-018-0482-1

[12] J. P. Cohen, J. D. Viviano, P. Bertin, P. Morrison, P. Torabian, M. Guarrera, M. P. Lungren, A. Chaudhari, R. Brooks, M. Hashir, and H. Bertrand, "TorchXRayVision: A library of chest X-ray datasets and models," *https://github.com/mlmed/torchxrayvision*, 2020. [Online]. Available: https://github.com/mlmed/torchxrayvision

[13] C. Nagpal, W. Potosnak, and A. Dubrawski, "auton-survival: an open-source package for regression, counterfactual estimation, evaluation and phenotyping with censored time-to-event data," 2022.

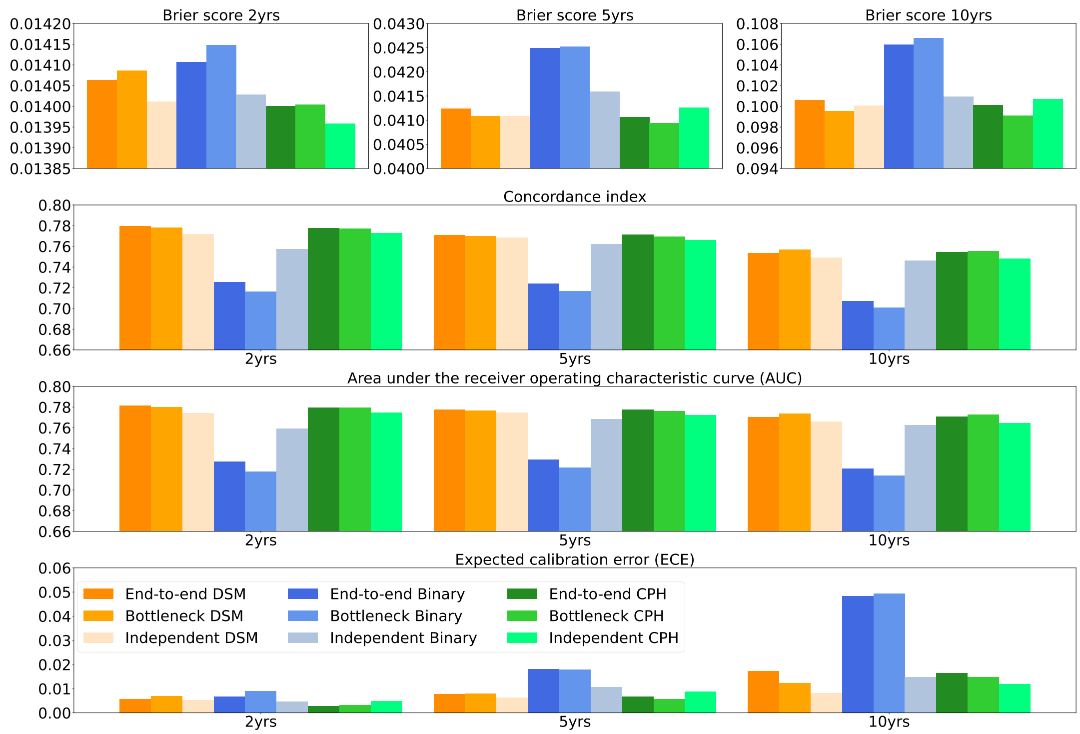


Fig. 5: Joint and independent models

[14] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang, "Concept bottleneck models," *CoRR*, vol. abs/2007.04612, 2020. [Online]. Available: https://arxiv.org/abs/2007.04612

[15] G. L. Andriole, E. D. Crawford, R. L. r. Grubb, S. S. Buys, D. Chia, T. R. Church, M. N. Fouad, C. Isaacs, P. A. Kvale, D. J. Reding, J. L. Weissfeld, L. A. Yokochi, B. O'Brien, L. R. Ragard, J. D. Clapp, J. M. Rathmell, T. L. Riley, A. W. Hsing, G. Izmirlian, P. F. Pinsky, B. S. Kramer, A. B. Miller, J. K. Gohagan, and P. C. Prorok, "Prostate cancer screening in the randomized prostate, lung, colorectal, and ovarian cancer screening trial: mortality results after 13 years of follow-up." *J Natl Cancer Inst*, vol. 104, no. 2, pp. 125–132, Jan 2012.

[16] H. Uno, T. Cai, M. J. Pencina, R. B. D'Agostino, and L. J. Wei, "On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data." *Stat Med*, vol. 30, no. 10, pp. 1105–1117, May 2011.

[17] C. Nagpal, M. Goswami, K. Dufendach, and A. Dubrawski, "Counterfactual phenotyping with censored time-to-events," *arXiv preprint arXiv:2202.11089*, 2022.

Less Is More: A Robust Visual Inertial Odometry with Active Feature Selection

Muhan Lin[1], Shibo Zhao[2] and Sebastian Scherer[2]

***Abstract*— It is a usual practice for visual odometry and SLAM to track several hundreds of points in real time. Although this practice has a good performance on high-end desktop PCs, it is difficult to apply this practice to some mobile platforms with limited computation resources, such as VR, Micro UAV and multi-camera system. Additionally, noisy points may decrease the accuracy of visual odometry and SLAM. Therefore, fewer but more informative features can boost both efficiency and accuracy. In order to solve this problem, we propose a new criterion for the active feature selection of VIO and then incorporate this method into an advanced VIO system, TP-TIO [1]. The new system was tested in the mmpug datasets [2], which were extracted in a long and dark corridor. The residuals can be reduced to 56.23% of the ones generated by the original TP-TIO without increasing the processing time and CPU usage of tracking, which is visualized in Fig 1.**

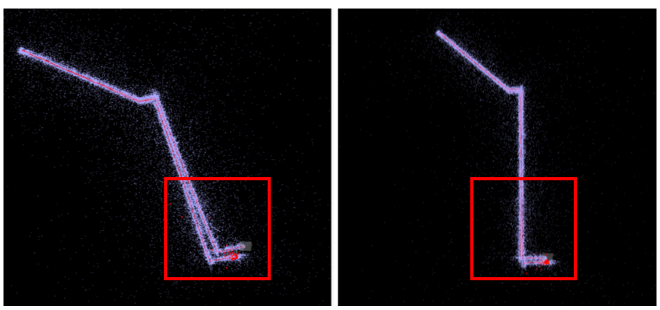


Fig. 1. The estimated trajectory without and with the active feature selection (threshold = 700). The vehicle starts from the starting point, goes for a long distance, goes back on the same route and returns to the starting point. Large drifts are shown in the picture above, which is generated by the original TP-TIO, while the drifts are very small in the picture below, which is generated by the improved TP-TIO. The noisy point cloud is also cleared to a large degree.

***Index Terms*— Computer Vision for Automation, Localization, Mapping**

I. INTRODUCTION

Visual inertial odometry (VIO) works as an effective method of estimating poses with high accuracy. It estimates states by integrating the feature tracking on camera images and the motion information from the Inertial Measurement Unit (IMU). VIO generally consists of the front end, which detects and tracks feature points, and the back end, which is responsible for mapping and pose estimation. The typical VIO works include VINS-mono [3] and the ORB-SLAM series [4] [5] [6] which show high accuracy.

However, when the algorithms are tested in some challenging environments of TUM-VI Benchmark [7], the residuals can increase dramatically, implying the lack of robustness. One reason of this problem is that more flawed or uninformative features are involved in the estimation in challenging environments. The noisy points can stain the estimation results of VIO. In addition, the large number of features makes the VIO systems hard to be applied to micro mobile equipment, which lacks high-end CPU and GPU to cope with the high computational demands. Therefore, we desire to choose features actively according to the quality of the features.

This work proposes a new feature selection method for VIO tracking based on quantifying intensity changes around the feature point [8], as intensity changes can render the confidence of features. This work then incorporates this method into TP-TIO, leveraging the advantages of both selection and adding points. The complete workflow chart of the new system is proposed. This algorithm is tested in the mmpug datasets [2], which were collected in a dark and long corridor. We changed the selection threshold and analyzed the algorithm in accuracy, the tracking time per frame, the CPU usage of tracking, and the number of filtered points. The estimation residual can be decreased to 56.23% of that from the original TP-TIO without increasing the processing time and CPU usage of tracking, increasing the robustness considerably. It is visualized by Fig 1.

II. RELATED WORK

A. Efficient Feature Extraction and Tracking

The front-end of VIO need to sample image points, detect features and track all feature points. The enormous computation complexity contributes to the trade-off between processing speed and CPU usage. [9] proposes a novel map simplification and a decoupled back-end optimization method, significantly reducing the computation complexity. The speed is increased to a large degree, whereas its CPU utilization sometimes is relatively high, which is not less than that of VINS-mono, and ORB-SLAM [10]. To cope with this problem of high CPU usage, VINS-mono restricts the number of tracked feature points, but this may sacrifice the accuracy of pose estimation and make the system less scalable for the multi-camera group. Another popular solution is to adopt the graphics processing unit (GPU). There are some concerning mature libraries, including Vision Programming Interface (VPI) developed by NVIDIA [11]

[1]Muhan Lin is with the Department of Electronic and Information Engineering, Computer Engineering, The Chinese University of Hongkong, Shenzhen, China. 119010177@link.cuhk.edu.cn
[2]Shibo Zhao and Sebastian Scherer is with the Air Lab, Carnegie Mellon University, Pittsburgh, PA 15213 shiboz,basti@andrew.cmu.edu

and CUDA Visual Library (VILIB) [12]. However, this may increase the hardware requirements and costs for vehicles, thus shrinking the application scenarios.

B. Robust VIO

Extracting more informative points is a good way to increase robustness of VIO. [13] supports the multi-camera system to avoid the accidental poor performance of one camera. However, the method may increase the computation amount. The reasons are that the pose estimator needs to process each pixel of each keyframe to get features and map points in the front end and the triangulation, and a large number of feature points leads to a large map and the large size of the matrices solved by the optimization algorithm in the back end.

TP-TIO [1] is a robust thermal inertial odometry (TIO) system integrating deep learning and TIO. This system can also be used as VIO because of the same working theory. Although TIO has a more stable performance than VIO in visual-degraded environments, the gradients of intensity in thermal images are too small and difficult to process. The deep learning method proposed in TP-TIO overcomes this problem by extracting more informative features and makes the system more robust. The resulted feature selection scheme gets influenced by the photometric changes least. Nonetheless, this system makes a trade-off between robustness and the computation amount, as a considerable number of features need to be extracted in the deep learning stage. After selecting the most informative features, the computation amount can be reduced without sacrificing robustness by using smaller but more informative feature sets.

C. Active Feature Extraction

The basic idea of active feature selection is to give each feature a weight, which will be small if the point cannot offer much effective information, like the features extracted from white walls. In this way, useless features and flawed features which can be caused by broken cameras and excessive exposure can be filtered. Covariance propagation is the fundamental way to weight the confidence of features, whose mathematical support is offered by [14]. Based on this concept, several works on feature selection have been published. [15] proposes the Good Line Cutting method for selecting good line features, but this may not be robust enough for environments like grasslands. [16] formulates weights for the estimated relative poses, which will not be directly involved in weighting features. [17] weights features with stochastic gradients and incorporates the weights of all features in optimization. Nevertheless, this method does not filter out the flawed points. Therefore, it may not be able to reduce the CPU usage of the whole system significantly. Additionally, the pose estimation residuals caused by the mistaken feature points may only be decreased but not wiped out. The hybrid VIO [18] and the Good Feature Matching [19], based on which [20] is derived, weight each feature points and select the best feature subset for local mapping. [18] uses the tracking length of each feature as their weights, but the tracking length is influenced by the motion route. Therefore, even if the vehicle observes the same point in the same direction and at the same distance, the weight derived by the hybrid VIO can differ as long as the route is distinct. The Good Feature Matching computes the feature uncertainty based on the reprojection errors. However, the existing VIO system involves several optimization strategies to reduce reprojection errors. The feature weights designed to be computed in the front end may not be the actual weights after some optimization in the back end.

III. METHODOLOGY

A. The Active Feature Selection

This work utilizes the ThermalPoint method to detect features and selects features out of the results, as this method gets influenced by the photometric changes least. It can thereby extract more stable features. The selection criterion is developed from the analysis of the intensity changing speed of features in [8], which is stated here first. The basic idea is to utilize the intensity gradient with respect to the x and y direction. We format M as the covariance matrix of intensity gradients, and then get its singular values λ_1, λ_2 by the Singular Value Decomposition.

$$I_x = \frac{\partial I}{\partial x} \tag{1a}$$

$$I_y = \frac{\partial I}{\partial y} \tag{1b}$$

$$M = \sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{1c}$$

$$= R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R \tag{1d}$$

The maximum singular value λ_{max} quantifies the largest intensity differences between the current point and the pixels around it. The ellipse centered at the analyzed feature point in Fig. 2 visualizes the intensity changing speeds at this point in all directions. The intensity of all the points on the ellipse's edge is considered the same. The lengths of the half axes are the path lengths for which the point needs to go to have the intensity on edge. Therefore, the two halves of the short axis represent the direction in which the intensity changes fastest, while the two halves of the long axis are the direction of changing slowest. The size of the short half axis should be composed by $(\lambda_{max})^{-0.5}$, the inverse of the most significant changing speed. Similarly, the size of the long half axis is related to $(\lambda_{min})^{-0.5}$. (To determine the uncertainty, only the scale but not the speed direction is considered at this point, so we always mean the absolute values of singular values by default when we mention λ in this paper.) We propose to quantify the uncertainty with $(\lambda_{max})^{-0.5}$, the length of the short half axis. The uncertainty degree can thereby be visualized as circles with a radius of $(\lambda_{max})^{-0.5}$. Large circles stand for great uncertainty. The reason why we choose the maximum singular value but not the minimum one is shown in Fig. 3. The red point in the center of this patch is the analyzed one. The intensity changing speed in the

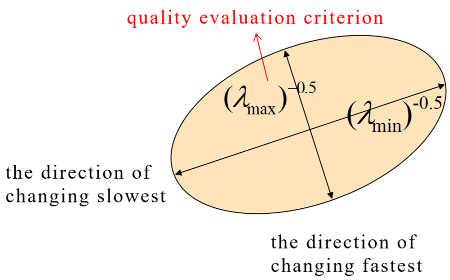


Fig. 2. The ellipse visualizing the intensity changing speeds at the analyzed point in all directions

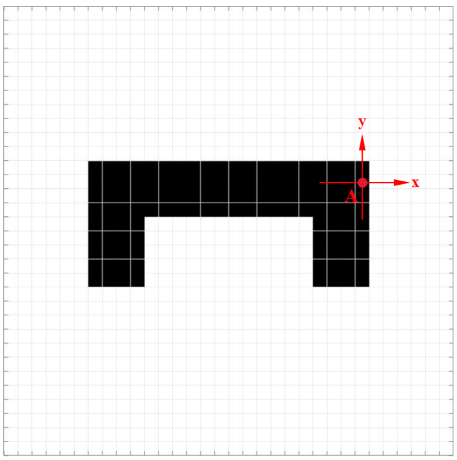


Fig. 3.

y direction I_y is always 0, thus always being the minimum. The minimum singular value hence is 0. However, I_x is of a considerable scale, making the maximum singular value considerably large. As a result, this edge feature with high confidence can be selected if selecting with the maximum singular value, while it will be ignored if selecting with the minimum singular value. For this sake, we should always use the largest intensity differences between the point and the pixels around it to evaluate the confidence of this feature. When selecting points, we set a threshold, and all the points with $(\lambda_{max})^{-0.5}$ larger than they are filtered out. We have obtained the optimal threshold for $(\lambda_{max})^{-0.5}$, 750, after several experiments. The details are shown in the result part.

B. Incorporate the Active Feature Selection Scheme into TP-TIO

Fig. 4 is the overview of the new system proposed by this work. Our main contribution is the stages colored orange, which are merged with the baseline, TP-TIO. We first examine all stages of TP-TIO here to find a reasonable place for insertion. As shown by the flow chart in Fig. 4, the front end of TP-TIO mainly consists of three stages. After feeding a new infrared camera image into TP-TIO, IMU preintegration is executed, followed by photometric tracking if the image is not the first one. The system then extracts features with the deep learning module ThermalPoint, detecting and feeding the extra features into the photometric tracking stage. Meanwhile, the tracker sends the tracking results to the back end. The results are used in triangulation. The PnP method is then executed to obtain the relative poses of the consecutive frames. These poses are finally optimized with the Local Bundle Adjustment (LBA).

The feature detection should be the stage where the number of features increases dramatically. Therefore, the feature selection is expected to happen after this stage. The tracking and the triangulation need to process each feature point, so the number of features impacts the computation amount of these two stages. Additionally, these two stages involve the history image points into trajectory estimation. If more noisy points enter these stages, the estimation accuracy will decrease. It is thereby desirable to have the feature selection before them. Therefore, we should insert the active feature selection into the red loop, as shown in Fig. 4. Because the tracking stage abandons the points which are in the last frame but cannot be tracked from the current frame, we select features after tracking and right before the deep learning module and the triangulation. Otherwise, the system will waste time computing singular values for points that will be abandoned immediately. In addition, the system can decide the threshold for feature selection according to the current number of features if the feature filter is put at this position. It is to avoid excessive filtering in some cases that the features have become very few after abandoning points which cannot be tracked in the current frame. Based on all the analysis above, the complete workflow chart can be obtained, as shown in Fig. 4.

IV. RESULTS

The new system was tested in the mmpug [2] datasets, which were collected in a long and dark corridor, a challenging environment for VIO. In the dataset, the vehicle starts from the starting point, goes for a long distance, and reaches a point p after several turns. Then it goes back on the same route and returns to the starting point. Such a special route helps us evaluate the estimation accuracy by examining the drifts between the routes to and from p and the distance between the starting and ending points. All the tests were conducted on the computer Oryx Pro with the CPU Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz. We compared the performance of the original TP-TIO system and the new TP-TIO system with distinct thresholds for feature selection to evaluate the algorithm and choose the optimal threshold.

Fig. 5 shows the real-time outputs of the tracking with feature selection. The signs of the points tracked for more than ten frames are colored red, while other points are colored red. Crosses mark the rejected points, most of which are from the white walls, thus being less informative than others, as shown in Fig. 5. The algorithm successfully removes most of the useless feature points extracted from white walls. The accepted points are marked with circles, whose radius size is quantified with the ceiling of $\sqrt{\lambda_{max}^{-0.5}+1}$. The algorithm suggests that the radius size should increase as the uncertainty increases. It can be seen from Fig. 5 that the radii of edge points are relatively small and slightly larger than

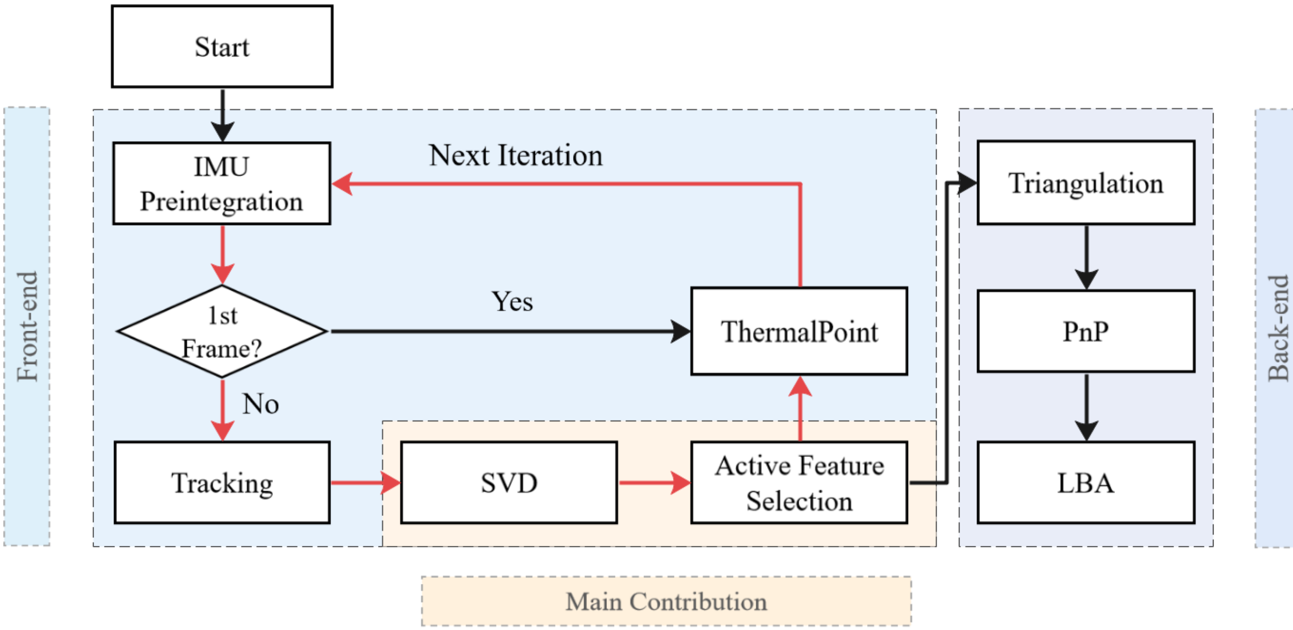


Fig. 4. The system overview. The TP-TIO is combined with active feature selection proposed by this work.

those of corner points. The points extracted from other places have large radii. This result is consistent with the fact that the features with low uncertainty are corner and edge points and the uncertainty of edge points is larger than that of edge points. Therefore, the active feature selection makes sense.

Fig. 5. The real-time output of the new system. The rejected points and the accepted points have been marked. The size of the circle radius represents the uncertainty of the accepted points

We tested the feature selection with thresholds of $\lambda_{max}^{-0.5}$ being 600, 650, 700, 750, and 800, respectively, and then compared it to the original TP-TIO. The comparisons are in terms of the residuals, the tracking time, the CPU usage of tracking, and the number of points filtered out. The results are plotted in Fig. 6. As we expected, more points were filtered out with the higher threshold. Additionally, the processing time and the CPU usage of tracking, which is also responsible for computing singular values and selecting points, have few changes with and without active feature selection. It implies that the additional computation on singular values and selection in the tracking part will not increase the computation burden. However, with the optimal threshold, which should be 750 from the experiment results shown in Fig. 6, the residual quantified by the distance between the starting and ending points can decrease to 56.23%, almost half of that of the original TP-TIO. In Fig. 1, we can see the considerable residual reduction implied by the slight drift between the route to and from p. The robustness has significantly increased. Additionally, it can be seen in Fig. 1 that the noisy point cloud around the estimated route is cleared to some considerable degree compared to the noisy point cloud generated in the original TP-TIO.

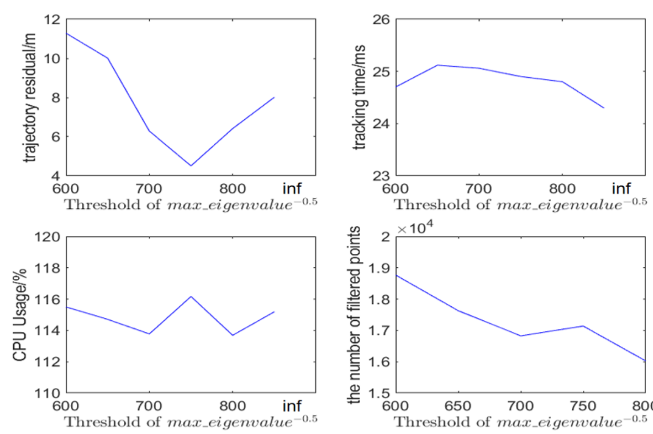


Fig. 6. The evaluation of the new system in terms of residuals, tracking time, CPU usage, and the number of filtered points

V. CONCLUSIONS

A robust information-driven VIO system based on active feature selection is realized in this work. We propose a new

< TABLE OF CONTENTS

feature selection criterion and incorporate it into the TP-TIO system to utilize the ThermalPoint, leveraging the strengths on increasing robustness of removing noisy and useless points and adding more informative points. In the experiments on challenging datasets, we successfully increased the accuracy and robustness. The optimal threshold for selecting features is also obtained in experiments.

VI. Future Work

The threshold of selection is given by users right now. Therefore, the performance of this selection now depends on the fixed threshold obtained from experiments. It may be helpful to use the adaptive threshold based on the current feature number, which can assist the selection algorithm in adapting to environments better.

Acknowledgment

This work is supported by the Robotics Institute Summer Scholars Program (RISS), Carnegie Mellon University, the Airlab, the Chinese University of Hong Kong, Shenzhen, and Shenzhen Institute of Artificial Intelligence and Robotics for Society.

References

[1] S. Zhao, P. Wang, H. Zhang, Z. Fang, and S. Scherer, "Tp-tio: A robust thermal-inertial odometry with deep thermalpoint," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4505–4512.

[2] S. Scherer, M. J. Travers, K. Cujic, D. Kohanbash, N. Shoemaker-Trejo, R. Darnley, and J. Spisak, "Mmpug: Multi-model perception uber good," 2022, accessed 4 August 2022. https://www.ri.cmu.edu/project/mmpug-multi-model-perception-uber-good/.

[3] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[4] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[5] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[6] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[7] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "The TUM VI benchmark for evaluating visual-inertial odometry," *CoRR*, vol. abs/1804.06120, 2018. [Online]. Available: http://arxiv.org/abs/1804.06120

[8] K. Kitani, "Harris corners," 2017.

[9] D. Rückert and M. Stamminger, "Snake-slam: Efficient global visual inertial slam using decoupled nonlinear optimization," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021, pp. 219–228.

[10] J. Jeon, S. Jung, E. Lee, D. Choi, and H. Myung, "Run your visual-inertial odometry on nvidia jetson: Benchmark tests on a micro aerial vehicle," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5332–5339, 2021.

[11] NVIDIA. (2021) Vpi-vision programming interface documentation. Accessed: 2022-07-25. [Online]. Available: https://docs.nvidia.com/vpi/

[12] B. Nagy, P. Foehn, and D. Scaramuzza, "Faster than fast: Gpu-accelerated frontend for high-speed vio," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4361–4368.

[13] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "Svo: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.

[14] A. Fontan, L. Oliva, J. Civera, and R. Triebel, "Model for multi-view residual covariances based on perspective deformation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1960–1967, 2022.

[15] Y. Zhao and P. A. Vela, "Good line cutting: Towards accurate pose tracking of line-assisted vo/vslam," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 527–543.

[16] D. Muhle, L. Koestler, N. Demmel, F. Bernard, and D. Cremers, "The probabilistic normal epipolar constraint for frame-to-frame rotation optimization under uncertain feature positions," 04 2022.

[17] J. H. Jung, Y. Choe, and C. G. Park, "Photometric visual-inertial navigation with uncertainty-aware ensembles," *IEEE Transactions on Robotics*, pp. 1–14, 2022.

[18] O. Seiskari, P. Rantalankila, J. Kannala, J. Ylilammi, E. Rahtu, and A. Solin, "Hybvio: Pushing the limits of real-time visual-inertial odometry," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2022, pp. 701–710.

[19] Y. Zhao and P. A. Vela, "Good feature matching: toward accurate, robust vo/vslam with low latency," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 657–675, 2020.

[20] L. Zhang, D. Wisth, M. Camurri, and M. Fallon, "Balancing the budget: Feature selection and tracking for multi-camera visual-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1182–1189, 2022.

Learning Vehicle Dynamics through Interactions for Off-Road Driving

Parv Maheshwari[1], Samuel Triest[2], Wenshan Wang[2] and Sebastian Scherer[2]

***Abstract*— In practical tasks such as off-road driving, robots need to understand the physical properties of the world to deal with its complexity. The aim of our work is to enhance the performance of the vehicle model of a modified Yamaha Viking ATV, by learning from its interactions with an ever-changing environment. We show that this results in increased model fidelity in challenging scenarios such as loose soil, traversing over vegetation, etc. We perform traditional offline system identification for our vehicle model combined with two approaches for online system identification - a traditional approach and a novel learning-based method. We then move on to compare the accuracy of these vehicle models on real-world data.**

***Index Terms*— Model Learning for Control, Field Robots, Autonomous Vehicle Navigation**

I. INTRODUCTION

For off-road navigation, robots often have to perform aggressive maneuvers on rough terrain. Not only this, but the vehicle needs to adapt to changing environments and terrain. Hence there is a need for a robust and adaptive vehicle model which would allow the predicted future state of the robot to be as close as possible to the ground truth. In history, such an adaptive model is achieved by using system identification.

System identification aims to find a set of parameters (P) to best describe the vehicle model on the basis of given information. To the best of the authors' knowledge, even though there are no direct works on parameter estimation of vehicle models in the off-road driving domain, the past works show promising results of system identification in various other applications. These applications are not just restricted to on-road driving [1] as [2] leverages real-world data for modeling an industrial car-like tractor. There also exists use cases of system identification in both aerial [3] and underwater vehicles [3], [4]. In literature, mainly the works on system identification and parameter estimation of vehicle models can be categorized into offline and online approaches.

We formulate the problem of traditional system identification similar to [5] which encourages the use of traditional offline approaches like using the least squares methods to estimate the value of the unknown parameters. We use a gradient-based optimizer [6] to minimize our loss.

While the offline approaches have seemed to work fairly well in the past, to identify the parameters when no prior information is provided about them, some recent works like [7], [8] explicitly show the advantage of online approaches over using the offline approaches in real-life.

[1] Department of Mathematics at Indian Institute of Technology Kharagpur

[2] Robotics Institute at Carnegie Mellon University

Fig. 1. We perform system identification for a Customized Yamaha Viking ATV while traversing through various environments.

For traditional online system identification, we use the same methodology as its offline variant. Our motivation to continue using the least squares formulation with gradient descent for online system identification similar to the offline variant comes respectively from [9] and [10].

While the possibility and application of classical approaches have been well explored for system identification in both online and offline variants, on the other hand, learning-based approaches for system identification are quite uncommon. One of them is [11] which combines images from the front camera along with vehicle dynamics to learn the coefficient of friction that is used in the vehicle model. While this approach takes into account the future surroundings, it does not leverage the history of the vehicle's trajectory in any form, which can especially help a lot in determining the terrains that the vehicle has traversed on and most likely still traversing on.

With this motivation in this paper, we present a learning-based approach for online system identification which leverages the recycling history of the trajectory that the vehicle has already followed. This is done using a novel architecture for the neural network which uses the current parameters of the vehicle model along with a trajectory history to output an individual Gaussian distribution for each parameter. We further evaluate our approaches on real-world data collected similar to [12] using our testing platform as shown in Fig. 1. The result of this experiment shows that our learning-based approach is more adaptive and robust than both the online and offline variants of the traditional system identification.

The remainder of this paper is organized as follows. In Section II, we provide background on our vehicle model along with defining our aim. In Section III, we discuss the details of various approaches to System Identification. In Section IV, we present the description of our testing platform and explain about setting an experiment to measure the robustness of our approaches. In Section V we analyze the results of our experiment. Finally, in Section VI, we give concluding remarks and mention scopes for future work.

II. PROBLEM DESCRIPTION

Our primary focus is to design a vehicle model that accurately models the vehicle's dynamics taking into account its interactions with the real-world environment. We have used the 2 Wheel drive mode on the ATV which allows us to fundamentally formulate our vehicle modeling as a Kinematic Bicycle Model(KBM) as shown in Figure 2. Here, at i^{th} timestep, the KBM state (X) consists of X_i = [p_i^x, p_i^y, θ_i, v_i, δ_i] where p_i^x, p_i^y represent the position coordinates, θ_i is the yaw of the vehicle, v_i is the vehicle's velocity and δ_i is the steering angle. The control input (U) at the i^{th} timestep is represented by U_i which consists of [$\dot{v_i}$, $\dot{\delta_i}$] representing the acceleration and steering rate respectively. If the vehicle is moving at velocity v_i and rotating around an Instantaneous Centre of Rotation (ICR) with a steering of δ_i, the general vehicle dynamics can be represented as Equation 1.

$$f(X_i, U_i) = \dot{X_i} \;=\; \begin{bmatrix} v_i * cos\theta_i \\ v_i * sin\theta_i \\ (v_i \tan\delta_i)/L \\ \dot{v_i} \\ \dot{\delta_i} \end{bmatrix} \tag{1}$$

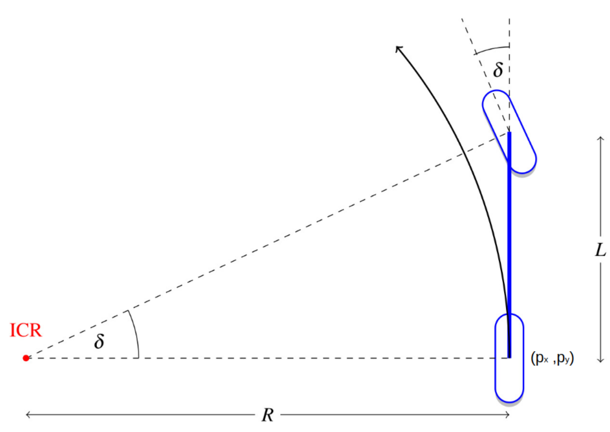


Fig. 2. Geometry of the bicycle model. The distance between the wheels is called wheelbase.

Now, at the i^{th} timestep, the next predicted state X_{i+1} can be denoted as $X_{i+1} = h(X_i, U_i)$ and can be calculated using combination of Equations 1 and 3 as f in Equation 2. For an improved estimate of the states, we use the 4th order Runge-Kutta method instead of Euler discretization.

$$h(X_i, U_i) = \;X_i \;+\; \frac{1}{6}(k_1 + 2k_2 + 3k_3 + k_4) \tag{2a}$$
$$k_1 \;=\; \Delta t * f(X_i, U_i) \tag{2b}$$
$$k_2 \;=\; \Delta t * f(X_i + \frac{k_1}{2}, U_i) \tag{2c}$$
$$k_3 \;=\; \Delta t * f(X_i + \frac{k_2}{2}, U_i) \tag{2d}$$
$$k_4 \;=\; \Delta t * f(X_i + k_3, U_i) \tag{2e}$$

where Δt is the resolution for the time step

Here we consider a modified version of the Kinematic Bicycle model where we provide throttle (T_i) and steering set point (δ_i^{target}) as actions. We define-

$$\begin{aligned} \dot{v_i} &= K_t * T_i - K_b * v_i - K_f \\ \dot{\delta_i} &= K_d * (\delta_i^{target} - \delta_i) \end{aligned} \tag{3}$$

where $P = (K_t, K_b, K_f, K_d)$ represents the set of parameters for our vehicle model. Here, K_t accounts for the effect of throttle on acceleration, $K_b * v_i$ is used to incorporate the engine braking of the vehicle as defined by [13], K_f represents the frictional force on the vehicle and K_d represents the proportional gain of the lower level steering controller. Our aim is to predict and estimate these parameters to increase the robustness and adaptive behavior of the vehicle model. These parameters can be estimated using system identification as further explained in Section III-B

III. METHODOLOGY

A. Data Collection

We have collected 30 minutes of off-road driving data in form of multiple discontinuous rosbags where each rosbag consists of multiple 5-second trajectories. This dataset aims to incorporate scenarios like acceleration, deceleration, turning, and special scenarios where the vehicle is traversing over vegetation and small rocks. These scenarios help us to find the right parameter as these cover different conditions where the effect of throttle, engine braking, and friction can influence the trajectory of the vehicle.

We have tried avoiding slopes while collecting data because as shown in Equation 3, $\dot{v_i}$ does not incorporate the effect of gravitational force, in the acceleration of the longitudinal velocity, which is non-negligible on slopes.

B. General System Identification

Here we estimate the classical vehicle model on the basis of the trajectory that the vehicle has followed. This is done by using a sequence of KBM states along with the actions as ground truth represented by $GT_{1:N}$ and $U_{1:N-1}$ respectively. Here GT_i and U_i represents the vehicle's current state and the commanded action at i^{th} timestep. Then the predicted trajectory ($S_{2:N}$) can be calculated as shown in Equation 4.

$$S_{2:N} = g(GT_1, U_{1:N-1}) \text{ where,}$$
$$g(GT_1, U_{1:N-1}) = \begin{bmatrix} h(GT_1, U_1) \\ h(h(GT_1, U_1), U_2) \\ \vdots \\ h(h(\ldots h(GT_1, U_1)\ldots), U_{N-1}) \end{bmatrix} \tag{4}$$

Given the ground truth and predicted trajectory, we calculate the loss as

$$\mathcal{L} = (GT_{2:N} - S_{2:N})^2 \tag{5}$$

We can perform system identification in two modes - offline or online.

C. Offline System Identification

We perform system identification to predict a set of values for the parameters in our vehicle model. This set can be represented as $P^o = (K_t^o, K_b^o, K_f^o, K_d^o)$ which in general can best describe the model. This is not done in real-time but rather performed on the dataset (Section III-A). For this mode, we minimize $\mathcal{L}$ to optimize the parameters set P by using the Adam optimizer [6]. The initial value of parameters can be arbitrarily set in this case.

D. Online System Identification

The online system identification similar to the offline system identification works on the history of the trajectory but unlike in the offline mode, the online mode uses real-time history to provide an updated set of parameters - $P^t = (K_t^t, K_b^t, K_f^t, K_d^t)$ at a time t. This is done because while P^o tries its best to represent the model in general, the online system identification works to provide history-specific parameters in real-time. For example, given the vehicle is traversing over pebbles and rocks, the frictional force which acts in the vehicle would be higher than what it would face while traversing over areas covered with vegetation. Hence the online system identification node would convey a higher value for K_f^t to the vehicle model than K_f^o. This helps us to increase the model accuracy in comparison to using a fixed set of parameters over different terrains and environments. The online system identification module conveys P^t to the vehicle model used by our local motion planning module as explained in Section III-F.

We have implemented Online system identification using two methods -

- Traditional Method - Similar to offline mode, we use [6] to minimize $\mathcal{L}$ for estimating P^t. Since the online system identification module has to update parameters in real-time, it is desirable for it to run at a frequency matching the frequency of the limiting observation. In our case, the limiting observation is the current position of the steering wheel which is received at 6 Hz. But due to the time taken in the forward and backward pass of a gradient-included rollout of a 5-second trajectory, in the current capacity, it is only possible to run the traditional method for a single epoch in real-time even after which this method can only run at 2 Hz.
- Learning Based Method - In this method, we use a neural network as shown in Figure 3 to predict the parameters when a history of trajectory and the current vehicle model parameters are fed into the network. Since our neural network can predict in real-time with almost little latency, we are able to use this method at a comparable rate to the limiting observation.

E. Learning-Based Online System Identification

1) Parameter Extraction: To train our architecture, we first extract out labels for P^t by using offline system identification on individual trajectories instead of using the entire collection of all the trajectories as done in the offline mode. To speed up the parameter extraction process while not hindering the accuracy of the extracted labels, we warm start the initial parameters for each trajectory with the final parameters of the last trajectory while using P^o as the initial parameters for the first trajectory for each individual rosbag.

2) Training of the architecture: As shown in Fig. 3, we represent the history of the trajectory represents the trajectory in the same KBM state space X, as explained in Section II, by processing a 5 seconds sequence of Odometry data combined with the position of the steering wheel. This KBM state history is first passed through a Wavenet encoder [14] which outputs a latent observation, which then is concatenated with the current parameters of the vehicle model. This concatenated input is passed through a multilayer perceptron (MLP) which outputs the mean and standard deviation of individual Gaussian distributions for the next set of parameters which are then selected by randomly sampling from the outputted distribution.

F. Local motion Control

We use MPPI [15] as a trajectory optimizer which provides us a local trajectory in form of a series of actions. We use our vehicle models to rollout sample trajectories in MPPI while optimizing for the following loss function -

$$\mathcal{J} = C(W_{pos}) + (\mathbb{1}_{v>=v_{v_{max}}})(e^{v-v_{max}} - 1) * (K_{penalty}) \tag{6}$$

where, W_{pos} is the position of the vehicle on the costmap, $C(p)$ is the value of the costmap at the p^{th} position, v_{max} is the maximum allowed velocity for the vehicle and $K_{penalty}$ is the speed penalty term usually kept as very high - for example - 10^8

IV. EXPERIMENTAL SETUP

A. ATV Platform

Similar to [12], various exteroceptive and Proprioceptive sensors were used to collect real-time data. We use a forward-facing Carnegie Robotics Multisense S21 stereo camera that provides us long-range high-resolution stereo RGB and depth images. For raw inertial measurements and estimates of position and velocity, a NovAtel PROPAK-V3-RT2i GNSS unit is used. As an addition to the sensors used in [12], we also use a forward-facing Velodyne LiDAR which provides us with laser scans of range up to 40m. All sensors and servos

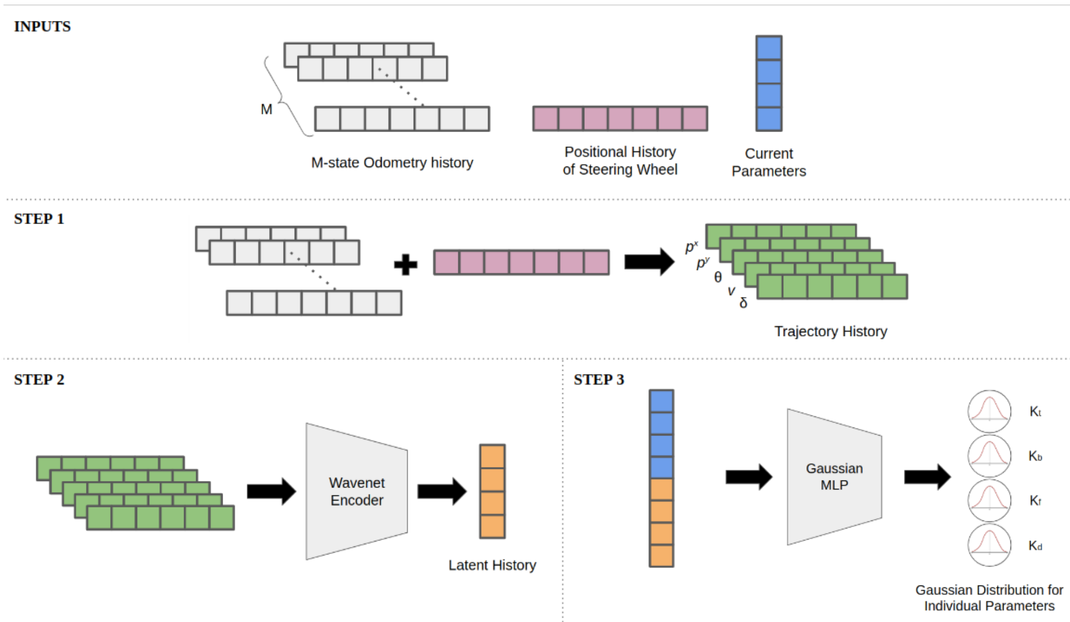


Fig. 3. Learning-based online system identification. We have used M=5.

were connected through ROS on an onboard computer. We have also relayed the joystick control inputs to the servos for the driver to manually operate the vehicle through a joystick. The sensors are integrated similar to Fig. 3 in [12].

Data from these sensors and servos are fed into our systems pipeline as shown in Fig. 4 to record the following data -

1) Robot Action: Actions $a = (\mu_t, \mu_s)$ were two-dimensional and corresponded to desired throttle and steering positions. Throttle commands took values between 0 and 1, with 1 corresponding to wide open throttle. Steering commands took values between-1 and 1, with -1 corresponding to a hard left turn. The commands were executed by the servos using PID position control.

2) Robot Pose: As an improvement to [12], instead of just using the raw measurements given by GNSS, we instead the raw measurements along with the laser scans, to run Super Odometry [16] which helps us achieve a more robust state estimation than the raw measurements from GNSS. We express the robot pose in the form of a concatenated position vector $p = (x, y, z)$, quaternion orientation $q = (q_x, q_y, q_z, q_w)$, linear velocity $v = (v_x, v_y, v_z)$ and angular velocity $w = (w_x, w_y, w_z)$. This is an improvement to [12] as we also consider linear and angular velocity and not just the position and the orientation vectors

3) Images: At each timestep, two RGB images were recorded from the onboard stereo camera.

4) Local Terrain Maps: : Similar to [12], we generate a local top-down view height map $M_h \in \mathbb{R}(w \times h \times 2)$ (two channels to represent the minimum height and maximum height) and a local RGB map $M_c \in \mathbb{R}(w \times h \times 3)$ using the stereo images from the Multisense S21 sensor and using the Stereo and Lidar Mapping Nodes. The cost maps generated from applying a lethal height threshold over the heightmaps maps are then used as explained in Section III-F.

B. Vehicle Model Accuracy

The vehicle models have been evaluated for their model accuracy on the data collected for system identification as mentioned in Section III-C. The performance has been measured in terms of the mean errors in all the individual elements in the KBM state and all of them combined. The results of this experiment have been reported in Table I.

V. RESULTS

As explained in Section III-D, to run the traditional online system identification in real time - we are only able to perform a single epoch of optimization over the previous labels. As expected this leads to a disturbance in loss $\mathcal{L}$ but since the offline optimization had already reached local minima to generally express the vehicle model, a single epoch results mostly in a downgrade of the performance rather than an improvement. This can be seen from Table I.

We further noticed that running many more optimization epochs (30 - 50) over an individual trajectory results in a P^t which is more accurate than P^o for that particular trajectory. With this motivation, we trained a neural network architecture and expected the performance, in general, of the vehicle model with Learning-based Online System Identification to be better than the other two models. This hypothesis is also confirmed from Table I.

VI. CONCLUSION AND FUTURE WORK

We present various methods of system identification along with a novel neural network to overcome the low-frequency output of the traditional online methods. We have also verified the accuracy of these models on real-world off-road data. Moving on, along with using the existing dataset,

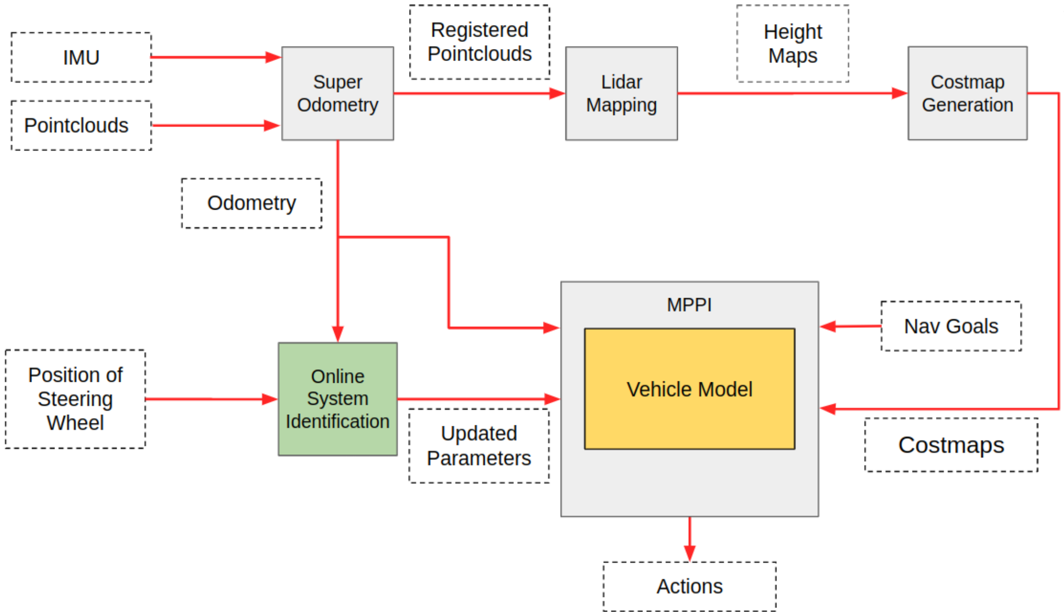


Fig. 4. Our complete Navigation stack. Here the dotted lines are only valid if the End-to-End learning-based vehicle model is used. **Remove if not explaining learning based model**

TABLE I
MEAN LOSSES FOR VARIOUS VEHICLE MODELS FOR INDIVIDUAL ELEMENTS IN THE STATE X AND ALL COMBINED

Model Type	L_{all}	L_x	L_y	L_θ	L_v	L_δ
KBM without Online System Identification	0.4285	1.4304	**0.1251**	0.0074	**0.5790**	0.0007
KBM with Traditional Online System Identification	0.5004	1.7164	0.1342	0.0075	0.6426	0.0011
KBM with Learning-based Online System Identification	**0.3189**	**0.8585**	0.1389	**0.0073**	0.5896	**0.0002**

we would also be using the entire TartanDrive Dataset [12] which is a dataset containing more than 5 hours of off-road driving data. This would not only help us achieve a more robust estimation of both the offline system identification and learning-based online system identification.

We are also motivated to incorporate additional input modalities like forward-facing terrain maps in our learning-based online system identification to also use the terrain features in the prediction of the model parameters. This would also shift the paradigm of the current online system to a more predictive-reactive approach rather than only reactive as it would use a map of the surroundings it has to traverse in the future while also using a history of states. Along with this future works can also incorporate the effect of gravity in our vehicle models. This would help our vehicle model to be more robust to changes in the pitch of the vehicle while it is traversing slopes.

ACKNOWLEDGMENT

This work was supported by the Robotics Institute Summer Scholars program at Carnegie Mellon University. The authors would like to thank Rachel Burcin and Dr. John Dolan for their support and organization of this program.

REFERENCES

[1] A. Ritter, F. Widmer, B. Vetterli, and C. H. Onder, “Optimization-based online estimation of vehicle mass and road grade: Theoretical analysis and experimental validation,” *Mechatronics*, 2021.

[2] H. Zhao, W. Chen, S. Zhou, and Y.-H. Liu, “Parameter estimation of an industrial car-like tractor,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4480–4487, 2021.

[3] A. Thondiyath, “Design, estimation of model parameters, and dynamical study of a hybrid aerial-underwater robot: Acutus,” pp. 423 – 430, 2019.

[4] A. Quraishi and A. Martinoli, “Online kinematic and dynamic parameter estimation for autonomous surface and underwater vehicles,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4374–4381.

[5] M. A. R. Pratama and A. Subiantoro, “Speed and yaw rate response optimization based on parameter estimation for electrical bus mathematical model,” in *2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)*, 2021, pp. 1–6.

[6] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2015.

[7] C. Böhm, C. Brommer, A. Hardt-Stremayr, and S. Weiss, “Combined system identification and state estimation for a quadrotor uav,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 585–591.

[8] M. Cooper, J. McGree, T. L. Molloy, and J. J. Ford, “Bayesian experimental design with application to dynamical vehicle models,” *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1844–1851, 2021.

[9] X. Yang, C. Wu, Y. He, X.-Y. Lu, and T. Chen, “A dynamic rollover prediction index of heavy-duty vehicles with a real-time parameter

estimation algorithm using nlms method," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 2734–2748, 2022.

[10] W. Chen, D. Tan, and L. Zhao, "Vehicle sideslip angle and road friction estimation using online gradient descent algorithm," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 11 475–11 485, 2018.

[11] D. Jin, B. Leng, X. Yang, L. Xiong, and Z. Yu, "Road friction estimation method based on fusion of machine vision and vehicle dynamics," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1771–1776.

[12] S. Triest, M. Sivaprakasam, S. J. Wang, W. Wang, A. M. Johnson, and S. A. Scherer, "Tartandrive: A large-scale dataset for learning off-road dynamics models," *ArXiv*, vol. abs/2205.01791, 2022.

[13] J. Mai, "System design, modelling, and control for an off-road autonomous ground vehicle," Master's thesis, Carnegie Mellon University, Pittsburgh, PA, August 2020.

[14] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *SSW*, 2016.

[15] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control using covariance variable importance sampling," *ArXiv*, vol. abs/1509.01149, 2015.

[16] S. Zhao, H. Zhang, P. Wang, L. Nogueira, and S. Scherer, "Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 8729–8736.

Blind Pedestrian Localization and Reorientation at Urban Crosswalks via UWB device

Rayna Hata[1], Isaac K. Isukapati[2], Zachary B. Rubinstein[2], Stephen F. Smith[2]

***Abstract*— Urban mobility and navigation through signalized intersections pose a challenge for vision impaired pedestrians. Constant vehicle noises, unclear crosswalk guides and cues, and the lack of sufficient time on the signal make crossing these intersections extremely difficult. PedPal, a mobile application, addresses these problems by personalizing the crossing experience for its users through real-time connectivity with the intersection's traffic control system. On one side, connectivity allows the user to communicate crossing intent to the intersection, and on the other PedPal keeps the user informed of the current crossing state. As the user crosses the intersection, PedPal utilizes Bluetooth beacons installed at each corner to track user progress and recognize when the crossing has been completed. Although the localization accuracy of these beacons has proven insufficient for more advanced tracking functions such as monitoring the user's position within the crosswalk, newer Ultra Wide Band (UWB) radio technology offers Centimeter-level accuracy, and provides new opportunities for enhancing the intersection crossing safety features of the PedPal app. In this paper we pursue this possibility. We assume that UWB beacons are installed at each corner of the intersection, and focus on the problem of detecting pedestrian movement outside of the crosswalk. Using distance measurements provided by the UWB beacons over time, we propose a localization and reorienting algorithm to ensure that the pedestrian will get to their destination safe and efficiently. A SUMO simulation model was developed to evaluate both algorithmic components. The localization method was found to provide high accuracy, differing by only 0.003% from the ground truth, and the reorienting algorithm was shown to consistently result in successful crossing within the crosswalk.**

***Index Terms*— Blind-Vision Impaired (BVI), Ultra Wide-Band (UWB),Urban Crosswalks, Blind-Pedestrian Navigation, Path Prediction**

I. Introduction

Walking as a method of transportation is often overlooked as being a challenge, yet for those in the blind/vision-impaired (BVI) community, walking, especially in urban settings, can prove to be a difficult task. A survey sent to 1,123 members of the Association for the Education and Rehabilitation (AER) of Blind's orientation and mobility team found that ninety-eight percent of respondents expressed that difficulty in knowing when to start crossing. Lack of aids such as audio cues or loud background noises interfere with the pedestrian's ability to know when the crossing signal turns green.Furthermore, in the survey, ninety-seven percent of their respondents stated that they had difficulty keeping straight path on a crosswalk. Sixty-six of the respondents of the survey said that they have trouble figuring out where the destination corner is. [1]

Vision-impaired pedestrians employ a range of methods to help cope with the challenges of crossing signalized intersections, including the use of a white cane, a guide dog, and/or cues learned through Orientation and Mobility (O&M) training (e.g., detecting the presence of textured concrete or gauging the sound of oncoming vehicles [2]). However, not everyone has access to a guide dog for reasons such as expenses, allergies, or availability, and the presence of other pedestrians, increased traffic noise, or other unpredictable factors complicate the use of various crossing cues in urban settings.

Additional physical infrastructure is also often added to intersections to provide crossing assistance to vision-impaired pedestrians. Such installments include noise emitting signals that alert the pedestrian when the signal is green and textured curb ramps to direct the pedestrian into the intersection. But, these installments cannot guarantee that the pedestrian will not drift out of the crosswalk as they move across the intersection. Furthermore, both pedestrians and O&M specialists find that looking for the curb ramp can cause more harm than good at times. While looking for the ramp, the pedestrian may get misaligned with the intersection and thus veer off in a different direction when crossing. [3]

Given the limitations of existing methods and infrastructure in supporting pedestrians with disabilities, research has turned attention in recent years to the use of smartphones as an assistive device. Through auditory [4] and haptic [5] feedback, text-to-speech features, and other accessibility features, it is felt that human-computer interactions will allow the vision-impaired community to more effectively and more safely navigate and explore the world. Several recent research efforts have focused specifically on the development of smartphone apps that exploit emerging "connected vehicle" technology to enhance pedestrian safety when crossing intersections. [6], [7], [8]. The *PedPal* safe intersection crossing app [8], [9], for example, has demonstrated that real-time connectivity can be used to personalize the pedestrian's crossing experience, allowing the user to communicate how much time is needed to safely cross to the traffic signal control system, and continuously using information received from the traffic control system to keep the user aware of the current crossing state. It has also demonstrated basic tracking capabilities as the user crosses, which until recently have been limited by localization accuracy issues.

In this paper, we consider the problem of extending

[1] Rayna Hata is with the department of Computer Science, Colby College, Waterville,ME USA rhata23@colby.edu

[2]Isaac Isukapati, Zachary Rubinstein, and Stephen Smith are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA (isaack,zbr,sfs)@cs.cmu.edu

PedPal's basic tracking capabilities to enable detection of and reaction to pedestrian movement outside of the crosswalk. Specifically, we propose a method of localizing and tracking pedestrian movement within the crosswalk through the use of ultra wide-band (UWB) radio beacons. UWB signals in outdoor line of sight (LOS) settings have an accuracy up to 2 to 3 cm [10], which offers tremendous sensing advantage over predecessor Bluetooth beacon and smartphone localization technologies, and is sufficient to accurately track a pedestrian. Using multiple UWB beacons positioned at each intersection corner and communicating with the PedPal app, we implement a pedestrian localization algorithm to triangulate the pedestrian's current location as well as determine if they are drifting away from the bounds of the crosswalk. If the algorithm determines that the pedestrian is drifting to the outer bounds, then a reorienting algorithm is applied to guide the pedestrian back to the center of the crosswalk to ensure safe crossing.

II. Background

A. PedPal

As indicated above, the starting point for our research is PedPal [8], [9], a smartphone app designed to promote safe intersection crossing for pedestrians with disabilities. PedPal utilizes emerging "connected vehicle" technology to communicate directly with the traffic control system at the intersection, and to actively influence its behavior to enhance safety. Most basically, PedPal knows the speed at which its user travels (e.g., using a white cane or a walker), and communicates this crossing constraint along with the user's chosen crossing direction to ensure that when the user receives the crossing signal, sufficient time will be allocated for crossing. Reciprocally, the app receives standard Dedicated Short Range Radio (DSRC) MAP and SPaT (Signal Timing and Phase) messages from the intersection to provide crossing options to the user and continual information about the current crossing state to prepare and inform the user about when it is time to cross. PedPal also actively monitors the crossing progress of its user through the intersection, and if it is detected that the user is crossing slower than expected (e.g., stumbles in a pothole and is slow getting up) then PedPal will request that the traffic signal system dynamically extend the green time to compensate. Pedpal also broadcasts its user's presence in the crosswalk to any approaching connected vehicles. Finally, for certain disabled pedestrians (such as motorized wheelchair users) PedPal can communicate expected arrival time information to the next intersection for purposes of streamlining intersection crossing time.

PedPal's functional capabilities are achieved by integrating the app with two distinct infrastructure technologies. First, PedPal is designed to inter-operate with Surtrac, a decentralized, real-time adaptive traffic signal control system [11], [12]. Within a Surtrac deployment, a processor is placed at each signalized intersection. At each intersection, a new traffic signal timing plan that minimizes the cumulative delay of all currently sensed traffic approaching the intersection is computed in real-time. Whenever a new timing plan is generated, the intersection then communicates to its downstream neighbors what traffic it expects to be sending their way (according to the plan), providing a basis for network-level coordination. Surtrac has been shown to lower average vehicle travel times by 25%, average number of stops by 30% and average amount of idle time by 40%. [11]. Surtrac is designed to incorporate multi-modal traffic flows, and as such, it is straightforward to factor pedestrian crossing demands into the signal timing plans that are generated.

The above mentioned capabilities provided by PedPal rely rather heavily on an ability to accurately determine when user's have arrived at a given intersection corner. In this regard, the native localization capability of the smartphone (in this case an Apple iPhone) has proved to be less than adequate. This has led to introduction of a second infrastructure technology, a set of fixed-position Bluetooth beacons located at each corner of the intersection. With this addition, it has been possible to use geo-fencing to provide the basic ability to recognize corners, and, for short crossing segments, this translates into some ability to track user crossing progress.

B. Movement to UWB beacons

The advent of UWB technology, however, has fundamentally changed the localization landscape. Some early experimentation with Apple AirTag technology indicated the potential for order of magnitude improvement in the ability to accurately determine object distances from the phone app, and more recent work with PedPal has utilized UWB technology to dramatically improve detection of arrival intersection corners and estimation of progress made toward crossing a particular street segment. For our initial experimentation in the field, a set of 4 Apple iPhones (Model 11 and above) have been utilized as initial UAB corner"beacons" at the intersection of Highland Avenue and Centre Avenue in the east end of Pittsburgh PA (see Figure 2). The ability to continuously exchange distance information between these UWB corner beacons and the PedPal app was established using Apple's Nearby Interaction framework[1], and initial experimentation has shown the ability to detect both arrival at intersection corners and the completion of crossing trips with a few centimeter accuracy up to a distance of 60 feet (the longest leg of the test intersections. Next steps will be to explore incorporation of 3rd party UAB beacons that are emerging on the market and support Nearby Interaction.

III. Detecting and Reacting to Movement Outside of the Crosswalk

Given the observed localization accuracy that is achievable with the use of UWB corner beacons, we consider their use in providing a more advanced tracking capability that relates directly to pedestrian safety, that of detecting and responding to pedestrian movement outside of the crosswalk. We can expect the MAP message received by the PedPal app from the intersection to provide the precise coordinates

[1] https://developer.apple.com/documentation/nearbyinteraction

of the crosswalk to be traversed. Thus, the challenge is one of situating the pedestrian within this physical space and prediction of the pedestrian's current trajectory. Accordingly, our solution is broken into two algorithmic components. Algorithm 1 involves localization of the pedestrian within the crosswalk. Algorithm 2 then predicts the pedestrian's future path, and, if it is determined to be drifting out of the crosswalk, the pedestrian is alerted to take corrective action.

Algorithm 1: Pedestrian Localization Process

$\vec{A} \leftarrow$ vector representing Geo locations of anchors
$d'_{i,\rho} \leftarrow$ distance between pedestrian and anchor i
$P'_{i,\rho} \leftarrow$ predicted Geo location of pedestrian w.r.t anchor i
$\vec{P'} \leftarrow$ vector representing predicted geo-locations of pedestrian from each anchor
while *The Crossing is incomplete* **do**
    Compute $\angle A$, $\angle B$ using $d, d_{1,\rho}, d_{2,\rho}$
    Estimate $P'_{i,\rho}$ from each anchor
    Set $\hat{P}$ as average of $P'_{1,\rho}$ & $P'_{2,\rho}$
    Call Path Prediction and Reorientation
end

Given a crossing episode between corner 1 and corner 2, Algorithm 1 periodically receives $d'_{1,\rho}$ and $d'_{2,\rho}$, the current distance readings between the pedestrian and each anchor respectively, throughout the pedestrian crossing. These distances along with the distance between the beacons (d) are used to geo-locate pedestrian ($P'_{i,\rho}$) with respect to each anchor. These calculations can be simply carried out using cosine rule as shown in Figure 1:

$$\cos(A) = (d^2 + {d'_{1,\rho}}^2 - {d'_{2,\rho}}^2)/(2 * d'_{1,\rho} * d'_{2,\rho}) \quad (1)$$

$$\cos(B) = (d^2 + {d'_{2,\rho}}^2 - {d'_{1,\rho}}^2)/(2 * d'_{1,\rho} * d'_{2,\rho}) \quad (2)$$

The final geo-location of the pedestrian is computed as the average of ($P'_{1,\rho}$, $P'_{2,\rho}$).

Next, we forecast the pedestrian's future path in order to determine if she is drifting out of the crosswalk. The heart of this algorithm is to determine and react quickly to the possibility of the pedestrian drifting out into the intersection. In order to do so, we must estimate and predict the next movements with reference to their initial pose estimate.

The pedestrian's initial pose estimation $\vec{O}^0_\rho$ is set as the reference coordinate against the succeeding pose estimations $\vec{O}^k_\rho$ to obtain the slope $m^j_{\rho,k}$ at step k. Step k indicates the number of readings taken in since the pedestrian started the crossing. The slope indicates the orientation that the pedestrian walks in relative to their original starting point at the beginning of the crossing.

If there is a change in the slope, it can be assumed that there is a change in the direction the person is walking in. However, it is necessary to compile multiple slope calculations to accurately predict if the pedestrian is drifting towards

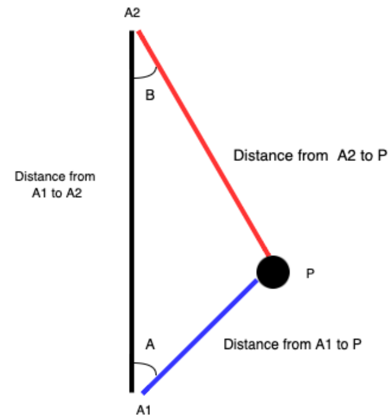


Fig. 1.   : This is a geometrical representation of the anchor and the pedestrian's location. The black line is the distance from anchor 1 to anchor 2. The blue line shows the distance between anchor 1 and the pedestrian, and the blue line shows the distance between anchor 2 and the pedestrian. In order to predict the pedestrian's location, we use angle 1 and angle 2 in equation 1 and 2.

the outer bounds of the crosswalk. Hence we use a heap queue Q to store j consecutive slope calculations. By taking the median value of Q, we can ensure that no outlier can throw off the prediction of the slope.

The median taken after collecting j slope calculations is subsequently used as a reference slope for determining if the pedestrian is straying. Thus, at each step k after the first j points, the oldest median is removed from the queue and the new slope between the pedestrian's initial pose estimation and the current pose estimation is added to the heap queue. After the slope is added, we compute the new median of the queue and compare that with the original median. It is important to use the reference slope as using any other slope will determine the pedestrian's direction from that slope.

If the new median is outside the threshold bounds, then there is indication that the pedestrian is not walking in a straight line towards the end of the crosswalk, and can no longer safely complete the crossing. If this movement continues, there is a high probability that pedestrian will leave the bounds of the crosswalk.

However, it is critical not to give instructions to the pedestrian until we can be sure that an outlier did not cause the reading. Therefore, we will continue to observe and not interfere until the pedestrian's slope is outside of the threshold for more than τ time steps. If that is true, then we start instructing the pedestrian to turn in the opposite direction of the slope. If the current median is larger than the threshold, then we will instruct the pedestrian to turn right. If the median is less than the threshold, then we will instruct the pedestrian to turn left.

The algorithm continues to localize and monitor the orientation of the pedestrian until the crossing has been completed. However, once the slope is within the threshold again, the algorithm determines that the pedestrian is back in the bounds of the crosswalk, and thus will stop giving instructions.

Algorithm 2: Path Prediction and Reorientation Process

$\vec{O}_\rho^0 \leftarrow$ pedestrian's initial pose estimation
$\vec{O}_\rho^k \leftarrow$ pedestrian's pose estimation at step k
$m_k \leftarrow$ slope of $(\vec{O}_\rho^k, \vec{O}_\rho^0)$
$Q \leftarrow$ heap queue containing slopes of last j steps
$m_\rho^0 \leftarrow$ initial pedestrian orientation (median of Q after first j steps)
$m_{\rho,k}^j \leftarrow$ pedestrian orientation at step k based on last j steps
$\vec{M}_t \leftarrow$ threshold bounds of median
$\tau \leftarrow$ consecutive number of steps $m_{\rho,k}^j$ is outside threshold bounds
$\tau_{max} \leftarrow$ max allowed number of steps for $m_{\rho,k}^j$ to be outside threshold bounds
At each step k do:
  compute m_k & update Q;
  set median of Q as $m_{\rho,k}^j$;
if $m_{\rho,k}^j \notin \vec{M}_t$ **then**
  τ++
end
if $\tau == \tau_{max}$ **then**
  start correction
end

IV. Evaluation

A. SUMO Simulation Framework

In order to visualize the pedestrian's movements and accurately reflect traffic conditions, the localization and path-planning algorithm is implemented in the Simulation of Urban Mobility (SUMO) software. SUMO is an open source continuous traffic simulation package that allows for the multi-modal simulation of urban traffic that displays the movements of vehicles and pedestrians on road structures imported via open-street maps. To simulate an intersection where the SURTRAC system is already deployed, we imported the intersection of South Highland Avenue and Centre Avenue in Pittsburgh, PA (see Figure 2).

Fig. 2. : A google earth image of Centre Ave. and Highland Ave., Pittsburgh,PA. In this image, Centre Ave is the horizontal line and Highland Ave. is the vertical line.

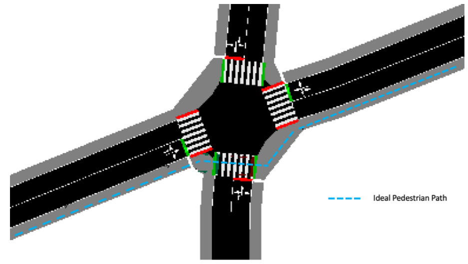


Fig. 3. : SUMO Visual of Pedestrian Path. The dotted line indicates the ideal pedestrian path.

Sumo's Traffic Control Interface (TracI) allows for manipulating pedestrians and vehicles as the simulation progresses. Using TracI, multiple pedestrian crossing scenarios with the pedestrian drifting toward the edge of the crosswalk were generated to mimic the potential movements of a blind pedestrian. As mentioned in the introduction, multiple factors, such as a reaction to traffic noise or an incorrect setup at the start of the intersection, can cause those in the BVI community to drift towards the edge of the crosswalk.

To manipulate and estimate the pedestrian's coordinates, our tracking procedure monitors and controls several parameters used by SUMO to drive the simulation, including pedestrian coordinates in (x,y) format, the angle that the pedestrian is facing, the pedestrian's speed, and the time step of the simulation (advancing in 100-millisecond intervals). Each of these parameters reflect data that would be accessible to PedPal in the field through its settings and underlying smartphone sensors.

Other crucial data streams that PedPal would be receiving from the UWB devices positioned at intersection corners in the field are the distances from each anchor beacon over time. As UWB device functions do not exist in SUMO, we manually compute what the smartphone would be receiving as distances from each of the UWB anchors. This is done by computing three euclidean distances at each point in time:

- $d_{1,\rho}$= Euclidean distance between anchor 1 and pedestrian
- $d_{2,\rho}$ = Euclidean distance between anchor 2 and pedestrian
- d = Euclidean distance between the UWB anchors

However, these distance calculations assume no signal noise and are thus not a realistic representative of readings that would be received in the field. Therefore, each distance between the anchors and the pedestrian is pushed through a noise model to emulate a real-world scenario.

The noise level added to the actual distance will increase or decrease as the pedestrian progresses along the intersection. This is reflective of the increase and decrease in noise as the smartphone gets further away or closer to a given UWB device. If the actual distance is more than half the length of the Euclidean distance between the two anchors, then a random δ_s of a range between 0.05-0.1m is added. If the distance is less, then a random δ_s of a range between 0.0-0.05m is added. The algorithm will take in the following new distances as the actual measurement between the pedestrian and the anchors, which is the equivalent of what the smartphone would be receiving as the distances between the pedestrian and the anchor in the field:

- $d'_{1,\rho}$ = Euclidean distance between anchor 1 and pedestrian
- $d'_{2,\rho}$ = Euclidean distance between anchor 2 and pedestrian

In the scenarios used in this simulation to evaluate the algorithm, the pedestrian starts the crossing in the correct direction but drifts towards the upper bounds of the crosswalk. This is used to emulate a scenario in which a factor such as noise or other pedestrians causes the person to start turning in the wrong direction. We determined the algorithm to be a success if it was able to do the following:

- Accurately predict the pedestrian's location with respect to the ground truth
- Detect when the pedestrian started to drift
- Reorient the pedestrian in the correct direction

B. Results

Using the localization method described in Section III, it can be seen that the our algorithm's estimation of the pedestrian's location is on target with the ground truth. Figure 4 shows the plot of each of the pedestrian location estimates. The plot of the average of estimated X, Y coordinates from A_1 and A_2 is plotted in yellow, and the ground truth coordinates are plotted in blue. The estimated X coordinate showed an average of 0.018% error from the ground truth, and the estimated Y coordinate showed an average of 0.027% error from the ground truth.

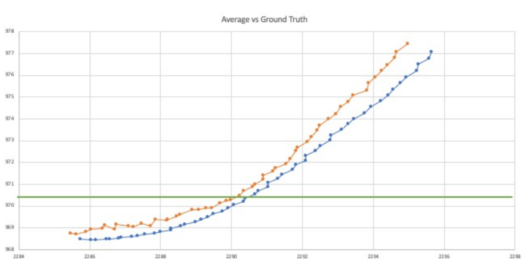


Fig. 4. : Plotting of the Coordinates. In this figure, the blue line shows the plot of the X,Y coordinates for the ground truth and the orange line shows the predicted X,Y coordinate of the pedestrian using the average of the two coordinate points from anchor 1 and anchor 2. The green line indicates the upper bound of the crosswalk and the plot shows the pedestrian's trip before the reorienting algorithm is applied. The path of the pedestrian has not been corrected in this graph.

The path predictor algorithm was shown to successfully predict when the pedestrian was drifting toward the upper or lower boundaries of the crosswalk, and in each case to then signal the pedestrian to move in the opposite direction back toward the middle of the crosswalk, until the pedestrian reaches an acceptable movement slope relative to the original slope. Figures 5 and 6 represent two boundary tests that we ran. In Figure 5, the orange path represents the complete trip of the pedestrian before the correction algorithm is run. In this scenario, the pedestrian is drifting towards the lower bound, and without correction, they will leave the crosswalk to the right. The blue line shows the pedestrian's path when the path prediction and reorientation are applied to their walk. Each point on the line indicates a step that the pedestrian takes. The orange and blue lines follow the same path until the algorithm detects that the pedestrian is drifting toward the lower bounds of the crosswalk. Upon detection, the algorithm signals the pedestrian to reorient, in this case sending single vibration signals to advise movement left and allow for a successful crossing. Figure 6 shows the alternative case where the pedestrian drifts towards the crosswalk's upper bound, and the pedestrian is directed (via two vibration signals) to move right to correct.

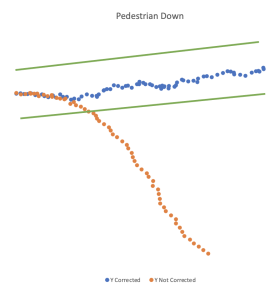


Fig. 5. Recognition and Correction of pedestrian drifting right

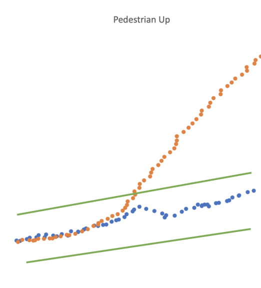


Fig. 6. Recognition and Correction of pedestrian drifting left.

V. Related Work

A. GPS tracking

GPS tracking is a common alternative to Bluetooth and UWB tracking. GPS tracking can potentially address the issue of determining if the pedestrian is out of the crosswalk's bounds. In one study, GPS readings are used to access the pedestrian's coordinates, and a satellite image of the location is pulled into a crosswalk detection framework to guide the user toward the intersection. While this method is helpful for real-time positioning of a pedestrian's location, the paper mentions that due to factors such as image acquisition problems or GPS accuracy, the localization accuracy cannot be narrowed down to cm. [13] Furthermore, this method assumes no ability for the mobile device to communicate directly with the traffic signal. Thus, while the satellite can guide the pedestrian to the desired crosswalk, it cannot notify the pedestrian of the state of the crosswalk, nor can it guide them during the crossing.

B. Indoor UWB Localization

UWB localization has been productively used in various settings, including the tracking of unmanned aerial vehicles [14], the tracking of objects in a factory environment to increase automation and efficiency [15], and the tracking of pedestrians indoors. One proposed method tracked a pedestrian in an indoor setting by exploiting a fusion of UWB sensors and inertial measurement units (IMU) sensors attached to the pedestrian's foot [16]. This method allows for error correction of the tracking measurements caused by non-line of sight instances. While this method effectively tracks the location of the pedestrian, it has not been tested in outdoor environments, and the number and placement of sensors required do not allow for seamless integration in the pedestrian's life.

C. Smartphone Based Tracking

Smartphones as assistive devices have increased over the years as smartphone capabilities have advanced. Applications such as BlindHelper [17] utilize smartphones to help blind pedestrians navigate their surroundings. BlindHelper is a smartphone-based application that takes location information, and traffic light status from the phone using GPS signals and outputs commands for the pedestrians to follow. The phone is connected via Bluetooth to a keypad that allows pedestrians to easily select their routes and a sonar distance meter to detect obstacles. This device allows for accurate tracking with an accuracy of up to 0.11m. However, the BlindHelper requires the pedestrian to hold several devices in their hand at once. This can often be inconvenient as their hands are not free to carry their belongings or hold anything.

D. Pedestrian to Infrastructure Communication

Pedestrian-to-infrastructure (P2I) communication is a crucial way to achieve connectivity between the smartphone and the signal. Smart Walk Assistant (SWA) [7] allows for WiFi communication between a smartphone and a roadside unit. The P2I connection implemented in SWA enables users to send a pedestrian signal request to the traffic signal controller and receive the traffic signal status. SWA also has a pedestrian-to-vehicle (P2V) communication that sends the location, speed, and heading of a vehicle to predict any conflicts between the pedestrian and approaching vehicles. PedPal similarly relies on P2I communication for optimization of signal timing plans at the intersection, but additionally employs UWB beacons to provide the types of safety applications considered in this paper.

VI. Conclusion and Next Steps

Precise localization is crucial when giving direction to those with visual impairments to obtain their trust in the assistive device. PedPal's goals are to localize the pedestrian, predict her path, and quickly respond to detected drift outside of the crosswalk during crossing.

Field tests conducted using UWB devices that have been mounted at corners of a signalized intersection have been shown to provide a reliable basis for localizing and tracking a smartphone being carried by an intersection crossing pedestrian. Given this assumption, we proposed an algorithm for detecting pedestrian movement outside of the crosswalk and for providing corrective advice when this movement is detected. An initial evaluation of the effectiveness of this algorithm was performed within a SUMO simulation environment. Results show that our algorithm is quite effective, and when compared to the ground truth coordinates measured in the simulation, prediction of pedestrian coordinates over time relative to known crosswalk boundary locations showed great accuracy. Our path prediction algorithm was able to successfully recognize when the pedestrian had moved outside of the crosswalk or was about to (depending on how the crosswalk boundary offsets were specified). Upon recognition of this circumstance, our reorientation algorithm was shown to provide helpful user advice without overwhelming the user.

Next steps include both generalizing the set of assumptions made in this initial evaluation and testing the approach in the field at real-world intersections. Our current Sumo simulation model moves the pedestrian, assuming they will start to drift out of the crosswalk in the middle of their trip. But that is not the only trajectory that might lead to a pedestrian moving outside of the bounds of the crosswalk. We want to also address scenarios where the pedestrian is not properly aligned and pointing in the right direction at the outset of crossing. In order to do so, we will need to incorporate the phone's inertial measurement unit (IMU) data into the pedestrian path prediction algorithm to understand their original orientation.

VII. Acknowledgements

The work reported in this paper was supported in part by the Robotics Institute Summer Scholars Program and its sponsors at Carnegie Mellon University as well as by National Science Foundation under award 2038612.

References

[1] B. L. Bentzen, J. M. Barlow, and L. Franck, "Addressing barriers to blind pedestrians at signalized intersections," *Institute of Transportation Engineers. ITE Journal*, vol. 70, no. 9, p. 32, 2000.

[2] "Travel tools and techniques of people who are blind or who have low vision." [Online]. Available: http://www.apsguide.org/chapter2_travel.cfm

[3] B. L. Bentzen, J. M. Barlow, and T. Bond, "Challenges of unfamiliar signalized intersections for pedestrians who are blind: Research on safety," *Transportation research record*, vol. 1878, no. 1, pp. 51–57, 2004.

[4] M. Banf, R. Mikalay, B. Watzke, and V. Blanz, "Picturesensation – a mobile application to help the blind explore the visual world through touch and sound," *Journal of Rehabilitation and Assistive Technologies Engineering*, vol. 3, p. 2055668316674582, Jan 2016.

[5] S. Khusro, B. Shah, I. Khan, and S. Rahman, "Haptic feedback to assist blind people in indoor environment using vibration patterns," *Sensors (Basel, Switzerland)*, vol. 22, no. 1, p. 361, Jan 2022.

[6] C.-F. Liao, "Using a smartphone application to support visually impaired pedestrians at signalized intersection crossings," *Journal of the Transportation Board*, vol. 2393, pp. 12–20, 2013.

[7] S. Khosravi, B. Beak, K. L. Head, and F. Saleem, "Assistive system to improve pedestrians' safety and mobility in a connected vehicle technology environment," *Transportation Research Record*, vol. 2672, no. 19, pp. 145–156, 2018. [Online]. Available: https://doi.org/10.1177/0361198118783598

[8] S. F. Smith, Z. B. Rubinstein, M. Dias, J. Marousek, and H. Radwanick, "Connecting pedestrians with disabilities to adaptive signal control for safe intersection crossing and enhanced mobility: Final report," Federal Highway Administration Joint Program Office Publication FHWA-JPO-19-754, 2019.

[9] G. Vadakpat, S. F. Smith, Z. B. Rubinstein, and M. B. Dias, "Technology to make signalized intersections safer for pedestrians with disabilities," *Public Roads*, vol. 84, no. 4, 2021.

[10] A. Jiménez and F. Seco, "Comparing decawave and bespoon uwb location systems: Indoor/outdoor performance analysis," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1–8.

[11] S. F. Smith, G. J. Barlow, X.-F. Xie, and Z. B. Rubinstein, "Smart urban signal networks: Initial application of the surtrac adaptive traffic signal control system," in *Twenty-third international conference on automated planning and scheduling*, 2013.

[12] S. F. Smith, "Smart infrastructure for future urban mobility," *AI Magazine*, vol. 41, no. 1, Winter 2020.

[13] M. C. Ghilardi, J. C. Jacques, and I. Manssour, "Crosswalk localization from low resolution satellite images to assist visually impaired people," *IEEE computer graphics and applications*, vol. 38, no. 1, pp. 30–46, 2016.

[14] F. Lazzari, A. Buffi, P. Nepa, and S. Lazzari, "Numerical investigation of an uwb localization technique for unmanned aerial vehicles in outdoor scenarios," *IEEE Sensors Journal*, vol. 17, no. 9, pp. 2896–2903, 2017.

[15] L. Barbieri, M. Brambilla, A. Trabattoni, S. Mervic, and M. Nicoli, "Uwb localization in a smart factory: Augmentation methods and experimental assessment," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–18, 2021.

[16] Z. Zeng, S. Liu, W. Wang, and L. Wang, "Infrastructure-free indoor pedestrian tracking based on foot mounted uwb/imu sensor fusion," in *2017 11th International Conference on Signal Processing and Communication Systems (ICSPCS)*, 2017, pp. 1–7.

[17] A. Meliones and C. Filios, "Blindhelper: A pedestrian navigation system for blinds and visually impaired," in *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2910674.2910721

Concept Learning for Interpretable Multi-Agent Reinforcement Learning

Renos Zabounidis, Joeseph Campbell, Simon Stepputtis, Dana Hughes, Katia Sycara
Carnegie Mellon University
{renosz, jacampbe, sstepput, danahugh, sycara} @andrew.cmu.edu

***Abstract*— Multi-agent robotic systems are increasingly operating in real-world environments in close proximity to humans, yet are largely controlled by policy models with inscrutable deep neural network representations. We introduce a method for incorporating interpretable concepts from a domain expert into models trained through multi-agent reinforcement learning, by requiring the model to first predict such concepts then utilize them for decision making. This allows a human operator to both query the resulting concept policy models to obtain a decision making rationale, as well as intervene in the event concepts are predicted incorrectly. We show that this yields improved interpretability and training stability, with benefits to policy performance and sample efficiency in a simulated and real-world cooperative-competitive multi-agent game.**

***Index Terms*— Multi-Agent Reinforcement Learning, Interpretable Machine Learning**

I. Introduction

With burgeoning adoption in fields such as autonomous driving, service robotics, and healthcare, multi-agent robotic systems are increasingly operating in real-world environments. The actions of these systems have a tangible and significant impact, particularly so when operating in close proximity to humans. While we expect such systems to exhibit safe and accurate behavior, errors are inevitable, and in such circumstances it is vitally important that the agents are able to explain their behavior to human operators. Operators can then ascertain whether the agent is operating erroneously – thus requiring intervention – or correctly but in a non-obvious manner.

However, state-of-the-art multi-agent systems are often controlled by deep neural network models trained with reinforcement learning techniques [1]. While these methods have shown great ability to generate effective and generalizable models, they do so at the expense of interpretability, and the models often remain inscrutable to human operators [2]. This poses a significant risk, especially in end-to-end models, where a rationale cannot be readily determined for *why* a model produced a certain decision, let alone provide a mechanism for intervening and correcting the rationale should it be incorrect. Such a mechanism is particularly important for robotic systems where we often encounter shifts in data distributions, such as when transferring policies from simulated environments to the real world, leading to model errors. These errors are exacerbated in multi-agent systems, where errors in each individual agent are compounded and produce large errors in environment dynamics.

In this paper, we propose a general method for learning interpretable policies – concept policy models – for multi-agent reinforcement learning (MARL). Our approach is predicated on the insight that we can leverage domain knowledge from an expert in order to regularize the model and influence what information is encoded from observations. We organize this domain knowledge into a set of interpretable *concepts* and enforce the constraint that the model is able to predict these concepts from observations, after which the concepts are used to predict policy actions. Concepts are semantically meaningful labels that can be extracted from observations, such as the presence of a concrete or abstract feature in an observation, e.g., the existence of a tree or the intention of a human. Crucially, we find that the regularization imposed by the concept information helps stabilize the training process, and as a result leads to improved performance and sample efficiency.

A typical end-to-end neural network policy model maps observations to actions. Our approach inserts an intermediate *concept* layer, as shown in Fig. 1 which is required to predict concepts from observations. While this yields an interpretable model [3], it also imposes the assumption that the set of concepts are sufficient for policy inference. To ease this constraint, we introduce a scalable *residual* layer which passes additional information to the subsequent policy layers while ensuring it remains decorrelated with the concepts. We posit that the interpretability of the model is proportional to the capacity of the residual layer; intuitively, the more residual information available, the less the model may rely on the concepts. We show that this can result in a trade-off between interpretability and accuracy, given the expressivity of the concepts. Our contributions are as follows, we

- Introduce a general method for learning concept policy models in MARL utilizing expert domain knowledge, enabling a human operator to understand a policy's decision rationale and improving accuracy, sample efficiency, and training stability.
- Develop two specific formulations based on this method: soft-concept models and hard-concept models, and empirically show the trade-off between accuracy and interpretability.
- Formulate an intervention methodology and show how this can be used to offset model errors in general and in transfer learning (sim-to-real) scenarios.
- Empirically show that our proposed approach produces interpretable, intervenable MARL policy models which exceed the accuracy of baseline MARL policies in a simulated and real-world game of "tag", between two

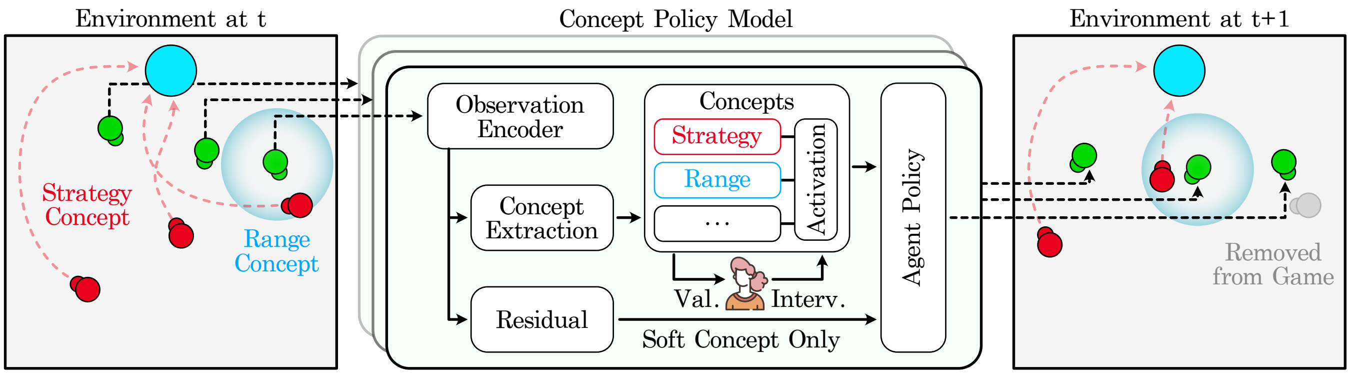


Fig. 1: Concept policy models predict a set of interpretable concepts from observations, which are then used along with an (optional) residual to predict a policy action. A domain expert may intervene and provide corrective concept values to the policy if mis-predicted.

teams of 2, 3, and 5 robots each.

II. Related Work

Interpretability in supervised learning: Intepretability has been extensively studied within the field of supervised learning [4], which can be largely grouped into two categories: the explicit creation of an intrinsically interpretable model, or the post-hoc transformation of an uninterpretable model to an interpretable one. The former case typically revolves around considering interpretable classes of models – decision trees [5], [6], linear models [7], [8], or rule-based methods [9], [10] for example – and developing algorithms for these models. In the latter case, uninterpretable models are either transformed into interpretable ones [11]–[13], or interpretable models are extracted from an uninterpretable model for the purpose of explaining a model's decision rationale [14], [15].

Concept models often fall into the transformation case, and have been studied within the context of transforming a set of uninterpretable feature embeddings into a set of interpretable concepts [3], [16], [17]. A recent approach [18] similarly uses concepts, but rather than directly predicting such concepts it attempts to align the internal model representation to coincide with them.

Interpretability in reinforcement learning: As in supervised learning, interpretability for reinforcement learning largely falls into the two categories of intrinsically interpretable models and post-hoc transformations. However, there is an additional line of work in which methods are devised to explain aspects particular to the Markov decision process model employed by RL methods. Some approaches have focused on interpretable representation learning [19], [20] and hierarchical decompositions [21], while others have opted to tackle MDP-specific explanations such as an interpretable reward signal [22] or action explanation [23], [24]. However, to the best of our knowledge, concept-related methods have not yet been explored in an RL setting, let alone MARL.

III. Preliminaries

Multi-Agent Reinforcement Learning: We model the MARL problem as a decentralized partially observable Markov decision process (Dec-POMDP) [25]. A Dec-POMDP is defined as a tuple $\langle \mathcal{S}, \mathcal{U}, P, R, \mathcal{Z}, O, n, \gamma \rangle$ in which $\mathcal{S}$ is the state space, $\mathcal{U}$ shared action space, P the state transition function, R the shared reward function, $\mathcal{Z}$ the observation space, O the observation function, n the number of agents, and γ the discount factor. For a given time step, the environment has a state $\mathbf{s} \in \mathcal{S}$ and each agent $a \in \{a_0, \ldots, a_n\}$ samples a partial observation $\mathbf{z}_a \in \mathcal{Z}$ according to the observation function $O(s, a) \in \mathcal{Z}$. The agents simultaneously sample an action $\mathbf{u}_a \in \mathcal{U}$ inducing a state transition according to $P(\mathbf{s}'|\mathbf{s}, \mathbf{u}) \in [0, 1]$. Each agent receives a reward r_a according to the shared reward function $R(\mathbf{s}_a, \mathbf{u}_a) \in \mathbb{R}$ with a discount factor $\gamma \in [0, 1]$. We follow a centralized training and decentralized execution approach (CTDE), thus learning a central policy $\pi_\theta(\mathbf{u}_a|\mathbf{z}_a) \in [0, 1]$ parameterized by θ by maximizing the discounted expected cumulative reward: $\mathbb{E}_t[\sum_t \gamma^t R(\mathbf{s}_a, \mathbf{u}_a)]$.

Multi-Agent Proximal Policy Optimization: Multi-Agent Proximal Policy Optimization (MAPPO) [26] is a straightforward extension to standard PPO [27] under the CTDE assumption in which we learn a single actor, π_θ, and a single critic, V_ϕ, parameterized by θ and ϕ respectively. When sampling from the environment, each agent executes the same learned policy with their individual observations and actions. As with all policy gradient methods, PPO seeks to compute the policy gradient by differentiating the following objective function:

$$L(\theta) = \hat{\mathbb{E}}_t[\log \pi_\theta(\mathbf{u}_a|\mathbf{z}_a)\hat{A}_t], \tag{1}$$

where $\hat{A}$ is the estimated advantage function. PPO extends this objective function by adaptively clipping the update gradient and applying an entropy bonus to the policy to encourage exploration. If the value function and policy function share parameters, i.e., $\theta = \phi$, then the objective function must also include the value function loss.

IV. Concept Policy Models

We propose a method for learning concept policy models, which integrates domain knowledge from an expert in the form of concepts into a neural network policy model. These concepts are intended to serve two purposes: they are useful predictors for the desired policy behavior and as such can serve as the basis of a decision rationale, and they can be corrected in real-time by an expert if mis-predicted by the policy in order to induce correct behavior. The expert provides an oracle function $V(\cdot)$ which can be used to predict a ground truth concept vector of size j given an observation from an agent a, $\mathbf{v} = V(\mathbf{z}_a)$ where $\mathbf{v} \in \mathbb{R}^j$. Concepts may be either continuous or discrete, and represent interpretable features which are assumed to be relevant to the task at hand. As an example, let us consider a cooperative-competitive multi-agent game in which two teams of agents play a game of "tag" during which one side must prevent the other from reaching a specific location. In this game, an expert might identify a specific feature such as the relative location of the nearest opposing team member, or a more abstract feature such as the opposing team's strategy as a concept.

A. *Policy Concept and Residual Layers*

We integrate this concept information into an end-to-end neural network policy model $\pi_\theta(\mathbf{u}_a|\mathbf{z}_a)$ which predicts the probability for agent a taking action $\mathbf{u}_a$ given the observation $\mathbf{z}_a$ and parameters θ. This is accomplished by inserting an intermediate layer $c_{\theta c}(\cdot)$ of size j into the network to estimate the concept vector, dividing the network into two parts: $\pi^1_{\theta 1}(\cdot)$ representing the portion of the network before the new layer, and $\pi^2_{\theta 2}(\cdot)$ representing the portion of the network after the new layer, such that

$$\pi(\mathbf{u}_a|\mathbf{z}_a) = \pi^2_{\theta 2}(c(\mathbf{x}) + r(\mathbf{x})) \quad \text{where} \quad \mathbf{x} = \pi^1_{\theta 1}(\mathbf{u}_a|\mathbf{z}_a) \tag{2}$$

and $r(\cdot)$ is a residual layer of size k designed to pass through non-concept information and the concept layer acts as a concept predictor, such that $\hat{v} = c(\mathbf{x})$. In our proposed concept policy model, $\pi^1_{\theta 1} : \mathbb{R}^{|\mathbf{z}|} \rightarrow \mathbb{R}^h$ acts as a feature encoder mapping an observation to a feature embedding. The newly inserted concept layer serves as a bottleneck such that $c(\cdot) : \mathbb{R}^h \rightarrow \mathbb{R}^j$ maps the feature embedding to a concept vector, while the residual layer $r(\cdot) : \mathbb{R}^h \rightarrow \mathbb{R}^k$ maps the feature embedding to a residual vector. The final policy layer $\pi^2_{\theta 2}(\cdot) : \mathbb{R}^{j+k} \rightarrow \mathbb{R}^{|\mathbf{u}|}$ maps the aggregated concept and residual vectors to a policy action. We train the concept layer $c(\cdot)$ by imposing an additional auxiliary loss $L^c(\theta)$ in the objective function optimized by MAPPO:

$$L(\theta) = \hat{\mathbb{E}}_t[\log \pi_\theta(\mathbf{u}_a|\mathbf{z}_a)\hat{A}_t] - L^c(\theta) \quad \text{where,}$$
$$L^c(\theta) = \sum_{i=0}^{j} L^c_i(\theta) \quad \text{and} \quad L^c_i(\theta) = \begin{cases} \mathrm{FL}(v_i, \hat{v}_i) & \text{if discrete} \\ \mathrm{MSE}(v_i, \hat{v}_i) & \text{if continuous.} \end{cases} \tag{3}$$

This loss is summed over each concept: mean squared error (MSE) is used for continuous concepts, and focal loss (FL) [28] for discrete concepts. The focal loss is a cross-entropy variant designed for class imbalanced situations which are likely to occur in our concept setting, as some concepts may be significantly rarer than others. In our above example with the strategy concepts, some strategies may be much less likely to occur than others, for instance. Note that for multi-class discrete concepts, a single abstract concept may consist of multiple nodes, and we refer to this as a *concept group*. In the strategy case, suppose an agent team may only execute one of strategy A, B, or C at a time, thus these three concepts represent a single concept group and so when we pass the discrete concepts through a softmax activation in order to calculate the focal loss we do so in a group-wise manner.

The goal of the residual layer is to pass through information from $\pi^1_{\theta 1}(\cdot)$ that is not captured by the concept vector. Without the residual, the concept vector must sufficiently represent the observation so that $\pi^2_{\theta 2}$ accurately infer the agent's action from concepts alone (a strict assumption in practice). We define two concept policy model variants: **hard concept policy models** which contain no residual ($k = 0$), and **soft concept policy models** which do ($k > 0$).

By examining the concept layer activations, a human operator can query the predicted concepts $\hat{v}$ and understand what concepts the policy model used for prediction. However, we conjecture that there is an inherent trade-off between the size of the residual layer k and the interpretability of these activations. While a full interpretability analysis is outside the cope of this work, we posit that the greater the residual dimension, the less that $\pi^2_{\theta 2}(\cdot)$ must rely on the concept vector, i.e., there is a larger amount of non-concept information on which to base its prediction – which follows that k is inversely proportional to interpretability.

B. *Concept and Residual Whitening*

In order to constrain the residual $r(\cdot)$ such that it does not encode information related to the concepts, we decorrelate the neuron activation vectors via whitening. Given a matrix $\mathbf{X} \in \mathbb{R}^{b \times j+k}$ consisting of the activations from the concatenated concept and residual vectors over a mini-batch of b samples, we aim to produce a whitened matrix $\mathbf{X}'$ with ZCA whitening via iterative normalization [29]

$$\mathbf{X}' = \mathbf{D}\Lambda^{-\frac{1}{2}}\mathbf{D}^T(\mathbf{X} - \mu_x) \tag{4}$$

where $\mathbf{D}$ and Λ are the eigenvectors and eigenvalues of $\mathbf{X}$ respectively. Iterative normalization uses an iterative optimization technique to incrementally whiten the matrix $\mathbf{X}$, where the hyperparameter T dictates the number of optimization iterations. This gives us the flexibility of only partially decorrelating the residual and whitening layers, if desired, by setting T to a smaller value, e.g., $T = 2$. In practice, we find that performing fewer iterations is often necessary to stabilize the training process, as a higher T tends to increase the stochastic normalization disturbance and leads to reduced training efficiency [29], which is particularly noticeable in a MARL setting. At each training iteration, we first perform whitening then backpropagate our computed gradients, thus allowing us to decorrelate the concept and residual layers without requiring an additional optimization step as in prior work [18].

C. *Policy Intervention*

In addition to querying the predicted concepts $\hat{v}$, a human operator may also decide to intervene when these predictions are incorrect. This can be achieved by explicitly overwriting the concept layer node activations (or softmax activations for discrete concepts) with the appropriate values. We denote the modified concepts as $\bar{v}$ which leads to the following intervened concept policy model: $\pi_\theta(\mathbf{u}_a|\mathbf{z}_a, \bar{v}) = \pi^2_{\theta 2}(\bar{v} + r(\mathbf{x}))$. This policy intervention corrects prediction errors in the feature encoder $\pi^1_{\theta 1}(\cdot)$. We find that these sorts of errors are particularly prevalent when transferring robot policies from simulation to the real world due to different observation distributions, and show that policy interventions are effective at reducing such errors in Sec. VI. Similar to interpretability, we hypothesize that intervention effectiveness is inversely proportional to the size of the residual k; we cannot intervene on the residual layer activations so any resulting errors will persist. While our empirical results hint in this direction, we save a full exploration for future work.

V. Experimental Setup

We show that our proposed concept policy models achieve high policy success rates and concept accuracy, in addition to improved training stability and sample efficiency, over standard MARL models in a cooperative-competitive multi-agent game of "tag" previously described in Sec. IV. We empirically analyze our approach in both simulated and real-world versions of this game, and explore its strengths and weaknesses especially with respect to interventions and sim-to-real transfer in Sec. VI.

A. Tag Game

In our game, two teams of agents compete with each other in which one team attempts to reach a specified goal location while the other team defends it and attempts to keep them away, which we refer to as the attacking and defending team respectively. An agent from each team may "tag" an agent from the opposing team as long as it lies within a given proximity and is facing the opposing agent, removing the tagged agent from play. The attacking team wins if any agent is able to reach the goal location, while the defending team wins if the attacking agents are all tagged or the maximum number of time steps elapses. Each team consists of the same number of agents, and we vary this number between 2, 3, and 5. This environment allows for complex behaviors in which attackers and defenders may employ coordinated strategies in order to out-maneuver the other team to win, and provides a suitable scenario for testing both specific and abstract concepts.

Observation and Action Space: The observations are a set of extracted features consisting of the positions, velocities, orientations and tagged status of all agents. Actions consist of accelerating forward or backward by a fixed amount, rotating left or right by a fixed offset, and tagging.

Reward: Our environment is a modified version of FortAttack [30]. We have simplified the reward function by removing penalties to encourage policy exploration, resulting in a reward of the form

$$R(s_a, u_a) = -R_{\text{Ori.}} - R_{\text{Miss}} - R_{\text{Tagged}} - R_{\text{Lose}} + R_{\text{Tag}} + R_{\text{Win}}, \quad (5)$$

where $R_{\text{Ori.}}$ is a penalty for not facing an opponent, R_{Miss} is a penalty for missing a tag, R_{Tagged} is a penalty for being tagged, R_{Lose} is a penalty for losing, R_{Tag} is a reward for tagging, and R_{Win} is a reward for winning. Several of these are shaping terms in order to improve sample efficiency, which we found to be necessary for efficient convergence rates in the absence of expert demonstrations.

Strategy: Furthermore, we restricted our game such that only the defending team's policy is trained via MARL. While it is a straightforward extension to train both an attacker and defender policy iteratively, we opted to restrict the attacking team to sampling strategies from a fixed policy distribution to better investigate the effects of our concept policy model on performance. We sample attacker strategies from a distribution consisting of three "types" with equal probabilities, {*random*, *left*, *right*}, where the attackers execute random actions, move towards the goal by sweeping along the left side of the environment, and move towards the goal by sweeping along the right, respectively. Given a sampled team level strategy, each agent then sampled an individual policy with noise from a further distribution, so as to generate stochastic policies.

Concepts: We utilized the following concepts: {*Range*, *Strategy*, *Target*, *Orientation*, *Position*}, in which *Range* is a boolean concept indicating whether the opposing agent specified by *Target* is within tagging range, *Strategy* is a categorical concept mapping to the above team-level strategies, *Target* is a categorical concept indicating an opposing agent that should be pursued, and *Orientation* and *Position* are continuous concepts encoding the relative orientation and position of each opposing agent, respectively. The hard concept policy models are trained with the full set of concepts, while the soft models only employ a subset consisting of {*Range*, *Strategy*, *Target*}.

Real-world Equivalent: The real-world version of our tag game is played in a 2v2 scenario on a $6' \times 6'$ play area, with four Khepera IV [31] robots. Policies are trained in the simulation environment, then executed in the real-world environment; no additional training and no few-shot conditioning is employed. The robot positions and orientations are extracted from a Vicon [32] motion capture system and converted into the model's expected observation format. The real-world version of the game exhibits significant differences in the dynamics between the simulated robots and the real-world robots – particularly in velocities, accelerations, and even control – presenting a challenging environment for sim-to-real. Further, tagged agents disappear in simulation while the real-world agents are driven out of the play area, providing a temporary obstacle.

B. Concept Policy Models and Baselines

We trained a hard and soft concept model for 10M time steps for each scenario – 2v2, 3v3, and 5v5 – along with a standard policy model without concepts. Each model consists of a series of fully connected layers, recurrent layers, and the iterative normalization layer applied over the concatenated concept and residual layers, with full details given in the supplementary material. The concept dimension j for each

	Setup	Model	WR	Intervened WR	*Range*	*Strategy*	*Target*	*Orientation*	*Position*
Simulation	2v2	Soft	51%	55%	0.03	0.04	0.24		
		Hard	83%	84%	0.04	0.07	0.2	0.10	0.11
		Base	34%						
	3v3	Soft	55%	57%	0.03	0.10	0.17		
		Hard	74%	80%	0.03	0.13	0.23	0.11	0.14
		Base	16%						
	5v5	Soft	32%	40%	0.02	0.25	0.48		
		Hard	78%	86%	0.03	0.14	0.13	0.11	0.21
		Base	31%						
Real	2v2	Soft	10%	0%	0.02	0.88	0.02		
		Hard	25%	95%	0.04	0.81	0.01	3.33	0.08
		Base	35%						

TABLE I: The win rate (WR) and concept errors for our proposed models (Soft and Hard) and a baseline without concepts (Base). The Hard model is trained over all concepts, the Soft model over a subset, and the Base model with none. The Win Rate is the standard win rate of the policy when the policy is executed, while the Intervened Win Rate (Intervened WR) is the win rate when an expert intervenes over all concepts. *Range*, *Strategy*, and *Target* are discrete concepts and as such the error shown is the error in accuracy score, while *Orientation* and *Position* are continuous and indicate mean squared error. *Orientation* is in radians and *Position* is a unit-less value in $[-1, 1]$.

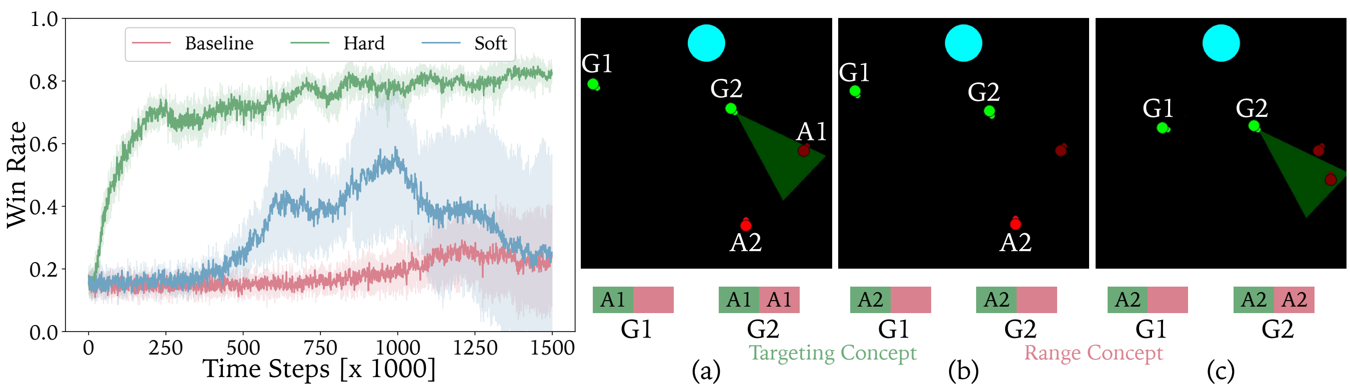


Fig. 2: Left: training curves showing win rate vs iterations over 5 random training seeds for each tested model type in a 2v2 scenario. Right: a sequence of episode steps showing the concept activations for agents on the defending team (green).

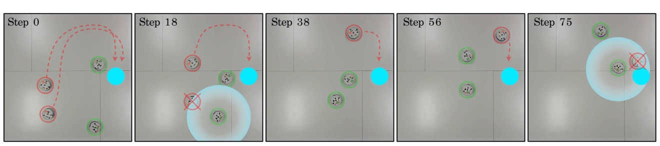


Fig. 3: A sequence of steps from a single episode during a policy execution in the real-world. The blue circle is the attacking team's (red) goal while the defending team (green) attempts to stop them.

hard model differ for each scenario due to the number of agents: $j = 13$, $j = 18$, and $j = 28$ for 2v2, 3v3, and 5v5 respectively. The concept dimensions for the soft models are $j = 9$, $j = 12$, and $j = 18$ for 2v2, 3v3, and 5v5. For the soft models, we additionally provide a residual layer with dimension $k = 23$, $k = 52$, $k = 78$ for 2v2, 3v3, and 5v5, respectively, leading to a combined bottleneck size of 32, 64, and 96. The baseline model lacks a concept layer ($j = 0$) and has a full-width residual $k = 128$. Residual layers sizes and other hyperparameters are given in the supplementary material and were chosen through extensive hyperparameter optimization.

VI. Results

The win rates and concept accuracy errors for the defending team in both simulation and real-world are shown in Table III. These values are computed by training two seeds with the best set of hyperparameters found during optimization, then rolling out each policy for 100 evaluation episodes in simulation, and 20 in real-world.

Simulation: We first observe that both concept policy model variants out-perform the baseline model in each scenario, with the hard concept model outperforming the others by a large margin. This in itself is unsurprising, given that the concepts were hand-designed so as to provide a sufficient amount of information for the policy, and the hard concept policy model heavily regularizes the learned model such that it learns this information. The decreased performance in the soft model is due to the fact that it is only trained with a subset of concepts, notably lacking both the *Orientation* and *Position*, and consequently the residual struggles to fully encode this information on its own. Due to the importance of these concepts, the performance suffers. The baseline model also significantly under-performs both of our proposed models, indicating that it struggles to learn appropriate features in the absence of regularization. The intervened win rate follows when an expert intervenes and sets the correct concept value when a concept is incorrectly predicted by the model. This results in an improved win rate, particularly in the 3v3 and 5v5 scenarios, which is reasonable given that these are more complex environments, and as can be seen in Table III, yield increased concept errors. Figure 2 shows the training curves for the three model types in a 2v2 scenario and clearly demonstrate the improved training stability offered by the hard concept model. Conversely, the soft concept models exhibit increased performance over the baseline models at the cost of decreased stability.

Real-world: In the real-world environment shown in Fig. 3, we can immediately see that the win rates are drastically reduced for the concept policy models, but surprisingly not for the baseline model. Qualitatively, we have observed that this is because the baseline model became trapped in a local minima and learned a policy which was semi-performant and independent of the actions of the opposing team. The baseline policy resulted in the defenders driving in circles around the goal while continuously tagging. This is clearly a sub-optimal behavior as it only reaches a 40% win rate in both simulation and the real-world, but it indicates the difficulty which standard MARL policies face when attempting to learn meaningful feature embeddings. The other interesting result from this experiment is the massive gain in performance by the hard model when interventions occur, and similarly the complete lack of improvement when the soft model is intervened. We can draw two insights from this: the distribution shift from the simulated to the real world environment is largely contained within the feature extractor, which is compensated for by the interventions in the hard model; and that the *Orientation* and *Position* concepts are by far the most important as when we are unable to intervene on them and correct for dynamics errors as in the soft model, performance fails to improve.

Concept Ablations: Next, we examine ablated hard concept policy models which are only trained over a subset of concepts in the 2v2 simulated scenario. The results are shown in Table II and further support the evidence that the *Orientation* and *Position* concepts are by far the most important with respect to the win rate. In the simulation environment, the win rate drops to 23% in the absence of those concepts, while in the real-world environment it drops to 27%. Note that the only difference between the hard model and the soft model with the RST concept set is the presence of a residual layer; when this residual is present and allowed to encode additional information the win rate is nearly doubled to 51% as in the soft model in Table III.

Intervention Ablations: We performed an ablation over the set of intervened concepts, with the results shown in Table II. We can first observe that ignoring interventions over the Orientation and Position concepts does not affect the win rate, likely because the associated errors for those concepts is already low as shown in Table III. As we intervene over fewer and fewer concepts, the win rate further drops; however, paradoxically the win rate drops to below the base win rate without any interventions at all. In the real-world we observe the opposite effect where intervening over a larger set of concepts always improves the win rate. This is especially so for the *Orientation* and *Position* concepts which account for a nearly 45% increase in the win rate alone, allowing us to conclude that not only are these concepts important, but that the distribution shift in the observation data from simulation to real particularly affects this concept.

Limitations: In order to analyze the performance of our model for a single team of agents, we have restricted the variability in our environment and reduced the complexity of possible behaviors. In the future, we would like to evaluate asymmetric team compositions and learn a policy for the attackers. We have also only considered low-dimensional inputs in our experiments, and although we expect our approach to scale well to rich input representations such as images, since concept models have been traditionally applied in vision domains, this remains an open question. Additionally, we would like to expand the complexity of our real-world environments by incorporating additional robots.

	2v2 Win Rate	3v3 Win Rate
All	83%	74%
RST	23%	27%
RTOP	80%	69%
OP	80%	72%

	Simulation Win Rate	Real Win Rate
All	84%	95%
RST	84%	20%
RTOP	81%	50%
OP	78%	80%

TABLE II: Top: Concept ablations in the 2v2 and 3v3 simulated scenario when only a subset of concepts are trained. Bottom: Intervention ablations in the 2v2 scenario when only a subset of concepts are intervened over. All consists of all concepts, RST = {*Range*, *Strategy*, *Target*}, RTOP = {*Range*, *Target*, *Ori.*, *Pos.*}, and OP = {*Ori.*, *Pos.*}.

VII. Conclusion

In this work we have introduced *concept policy models* for Multi-Agent Reinforcement Learning which incorporate domain knowledge from an expert in the form of concepts. We have developed a general framework in which concepts may be optionally augmented with residual information in order to ease the restriction that they fully express the information necessary for policy prediction, and show that this results in concept policy models which fall along a spectrum of regularization: hard concept models in which no residual is allowed, and soft concept models in which it is. We empirically show that this regularization greatly stabilizes training and results in improved accuracy and sample efficiency, and crucially, allows a human operator to query the model for its concept activations which provide an interpretable *rationale* for the policy's decisions. We further show that when the operator intervenes and corrects incorrect concept predictions, we can improve policy accuracy and partly compensate for distribution shifts, particularly in sim-to-real transfer scenarios.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[2] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[3] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang, "Concept bottleneck models," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5338–5348.

[4] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018.

[5] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens, "An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models," *Decision Support Systems*, vol. 51, no. 1, pp. 141–154, 2011.

[6] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[7] S. Haufe, F. Meinecke, K. Görgen, S. Dähne, J.-D. Haynes, B. Blankertz, and F. Bießmann, "On the interpretation of weight vectors of linear models in multivariate neuroimaging," *Neuroimage*, vol. 87, pp. 96–110, 2014.

[8] T. Le Nguyen, S. Gsponer, I. Ilie, M. O'Reilly, and G. Ifrim, "Interpretable time series classification using linear models and multi-resolution multi-domain symbolic representations," *Data mining and knowledge discovery*, vol. 33, no. 4, pp. 1183–1222, 2019.

[9] X. Yin and J. Han, "Cpar: Classification based on predictive association rules," in *Proceedings of the 2003 SIAM international conference on data mining*. SIAM, 2003, pp. 331–335.

[10] F. Wang and C. Rudin, "Falling rule lists," in *Artificial intelligence and statistics*. PMLR, 2015, pp. 1013–1022.

[11] R. Agarwal, L. Melnick, N. Frosst, X. Zhang, B. Lengerich, R. Caruana, and G. E. Hinton, "Neural additive models: Interpretable machine learning with neural nets," *Advances in Neural Information Processing Systems*, vol. 34, pp. 4699–4711, 2021.

[12] O. Bastani, C. Kim, and H. Bastani, "Interpretability via model extraction," *arXiv preprint arXiv:1706.09773*, 2017.

[13] G. Liu, O. Schulte, W. Zhu, and Q. Li, "Toward interpretable deep reinforcement learning with linear model u-trees," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 414–429.

[14] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti, "Local rule-based explanations of black box decision systems," *arXiv preprint arXiv:1805.10820*, 2018.

[15] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, *et al.*, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)," in *International conference on machine learning*. PMLR, 2018, pp. 2668–2677.

[16] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum, "Neural-symbolic vqa: Disentangling reasoning from vision and language understanding," *Advances in neural information processing systems*, vol. 31, 2018.

[17] M. Losch, M. Fritz, and B. Schiele, "Interpretability beyond classification output: Semantic bottleneck networks," *arXiv preprint arXiv:1907.10882*, 2019.

[18] Z. Chen, Y. Bei, and C. Rudin, "Concept whitening for interpretable image recognition," *Nature Machine Intelligence*, vol. 2, no. 12, pp. 772–782, 2020.

[19] T. Lesort, N. Díaz-Rodríguez, J.-F. Goudou, and D. Filliat, "State representation learning for control: An overview," *Neural Networks*, vol. 108, pp. 379–392, 2018.

[20] Z. Lin, K.-H. Lam, and A. Fern, "Contrastive explanations for reinforcement learning via embedded self predictions," *arXiv preprint arXiv:2010.05180*, 2020.

[21] B. Beyret, A. Shafti, and A. A. Faisal, "Dot-to-dot: Explainable hierarchical reinforcement learning for robotic manipulation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5014–5019.

[22] Z. Juozapaitis, A. Koul, A. Fern, M. Erwig, and F. Doshi-Velez, "Explainable reinforcement learning via reward decomposition," in *IJCAI/ECAI Workshop on explainable artificial intelligence*, 2019.

[23] F. Cruz, R. Dazeley, P. Vamplew, and I. Moreira, "Explainable robotic systems: Understanding goal-driven actions in a reinforcement learning scenario," *Neural Computing and Applications*, pp. 1–18, 2021.

[24] H. Yau, C. Russell, and S. Hadfield, "What did you think would happen? explaining agent behaviour through intended outcomes," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 375–18 386, 2020.

[25] F. A. Oliehoek and C. Amato, *A concise introduction to decentralized POMDPs*. Springer, 2016.

[26] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of mappo in cooperative, multi-agent games," *arXiv preprint arXiv:2103.01955*, 2021.

[27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[29] L. Huang, Y. Zhou, F. Zhu, L. Liu, and L. Shao, "Iterative normalization: Beyond standardization towards efficient whitening," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4874–4883.

[30] A. Deka and K. Sycara, "Natural emergence of heterogeneous strategies in artificially intelligent competitive teams," in *International Conference on Swarm Intelligence*. Springer, 2021, pp. 13–25.

[31] J. M. Soares, I. Navarro, and A. Martinoli, "The khepera iv mobile robot: performance evaluation, sensory data and software toolbox," in *Robot 2015: second Iberian robotics conference*. Springer, 2016, pp. 767–781.

[32] Vicon Motion Systems, "Vicon motion capture system." [Online]. Available: https://www.vicon.com/hardware/cameras/

[33] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, "A survey on neural network interpretability," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2021.

[34] J. Kleinberg and S. Mullainathan, "Simplicity creates inequity: implications for fairness, stereotypes, and interpretability," in *Proceedings of the 2019 ACM Conference on Economics and Computation*, 2019, pp. 807–808.

[35] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st international conference on neural information processing systems*, 2017, pp. 4768–4777.

[36] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine learning proceedings 1994*. Elsevier, 1994, pp. 157–163.

[37] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, K. Goldberg, J. Gonzalez, M. Jordan, and I. Stoica, "Rllib: Abstractions for distributed reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3053–3062.

[38] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu, "Reinforcement learning with unsupervised auxiliary tasks," *arXiv preprint arXiv:1611.05397*, 2016.

[39] E. Strumbelj and I. Kononenko, "An efficient explanation of individual classifications using game theory," *The Journal of Machine Learning Research*, vol. 11, pp. 1–18, 2010.

[40] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[41] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[42] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[43] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[44] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.*, "Mastering

chess and shogi by self-play with a general reinforcement learning algorithm," *arXiv preprint arXiv:1712.01815*, 2017.

[45] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in neural information processing systems*, 2000, pp. 1008–1014.

[46] M. Świechowski, K. Godlewski, B. Sawicki, and J. Mańdziuk, "Monte carlo tree search: A review of recent modifications and applications," *arXiv preprint arXiv:2103.04931*, 2021.

[47] J. Song, H. Ren, D. Sadigh, and S. Ermon, "Multi-agent generative adversarial imitation learning," *arXiv preprint arXiv:1807.09936*, 2018.

[48] A. Gopnik and H. M. Wellman, "Why the child's theory of mind really is a theory," 1992.

[49] N. Rabinowitz, F. Perbet, F. Song, C. Zhang, S. A. Eslami, and M. Botvinick, "Machine theory of mind," in *International conference on machine learning*. PMLR, 2018, pp. 4218–4227.

[50] M. Lanctot, V. Zambaldi, A. Gruslys, A. Lazaridou, K. Tuyls, J. Pérolat, D. Silver, and T. Graepel, "A unified game-theoretic approach to multiagent reinforcement learning," *arXiv preprint arXiv:1711.00832*, 2017.

[51] A. Lerer and A. Peysakhovich, "Learning existing social conventions via observationally augmented self-play," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 107–114.

[52] W. Yoshida, R. J. Dolan, and K. J. Friston, "Game theory of mind," *PLOS Computational Biology*, vol. 4, no. 12, pp. 1–14, 12 2008. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1000254

[53] C. Baker, R. Saxe, and J. Tenenbaum, "Bayesian theory of mind: Modeling joint belief-desire attribution," in *Proceedings of the annual meeting of the cognitive science society*, vol. 33, no. 33, 2011.

[54] C. L. Baker, J. Jara-Ettinger, R. Saxe, and J. B. Tenenbaum, "Rational quantitative attribution of beliefs, desires and percepts in human mentalizing," *Nature Human Behaviour*, vol. 1, no. 4, pp. 1–10, 2017.

[55] J. Tenenbaum, "Building machines that learn and think like people," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2018, pp. 5–5.

[56] J. Nash, "Non-cooperative games," *Annals of mathematics*, pp. 286–295, 1951.

[57] M. Liu, M. Zhou, W. Zhang, Y. Zhuang, J. Wang, W. Liu, and Y. Yu, "Multi-agent interactions modeling with correlated policies," *arXiv preprint arXiv:2001.03415*, 2020.

[58] Y. Wu, E. Mansimov, R. B. Grosse, S. Liao, and J. Ba, "Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation," *Advances in neural information processing systems*, vol. 30, pp. 5279–5288, 2017.

[59] J. N. Foerster, Y. M. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," *arXiv preprint arXiv:1605.06676*, 2016.

[60] C. Chen, O. Li, C. Tao, A. J. Barnett, J. Su, and C. Rudin, "This looks like that: deep learning for interpretable image recognition," *arXiv preprint arXiv:1806.10574*, 2018.

[61] P. Hernandez-Leal, B. Kartal, and M. E. Taylor, "A survey and critique of multiagent deep reinforcement learning," *Autonomous Agents and Multi-Agent Systems*, vol. 33, no. 6, pp. 750–797, 2019.

[62] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *International Conference on Autonomous Agents and Multiagent Systems*. Springer, 2017, pp. 66–83.

[63] S. Kumar, P. Shah, D. Hakkani-Tur, and L. Heck, "Federated control with hierarchical multi-agent deep reinforcement learning," *arXiv preprint arXiv:1712.08266*, 2017.

[64] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *arXiv preprint arXiv:1706.02275*, 2017.

[65] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2961–2970.

[66] H. He, J. Boyd-Graber, K. Kwok, and H. Daumé III, "Opponent modeling in deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1804–1813.

[67] G. Plumb, M. Al-Shedivat, Á. A. Cabrera, A. Perer, E. Xing, and A. Talwalkar, "Regularizing black-box models for improved interpretability," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[68] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144. [Online]. Available: https://doi.org/10.1145/2939672.2939778

[69] G. Plumb, D. Molitor, and A. S. Talwalkar, "Model agnostic supervised local explanations," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/b495ce63ede0f4efc9eec62cb947c162-Paper.pdf

[70] A. Anand, E. Racah, S. Ozair, Y. Bengio, M.-A. Côté, and R. D. Hjelm, "Unsupervised state representation learning in atari," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[71] T. Yang, Z. Meng, J. Hao, C. Zhang, Y. Zheng, and Z. Zheng, "Towards efficient detection and optimal response against sophisticated opponents," *arXiv preprint arXiv:1809.04240*, 2018.

[72] I. Oguntola, D. Hughes, and K. Sycara, "Deep interpretable models of theory of mind," in *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*. IEEE, 2021, pp. 657–664.

[73] L. Huang, D. Yang, B. Lang, and J. Deng, "Decorrelated batch normalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 791–800.

[74] L. Antwarg, R. M. Miller, B. Shapira, and L. Rokach, "Explaining anomalies detected by autoencoders using shap," *arXiv preprint arXiv:1903.02407*, 2019.

[75] J. B. Tenenbaum, "A bayesian framework for concept learning," Ph.D. dissertation, Massachusetts Institute of Technology, 1999.

[76] C. L. Baker, "Bayesian theory of mind: Modeling human reasoning about beliefs, desires, goals, and social relations," Ph.D. dissertation, Massachusetts Institute of Technology, 2012.

[77] N. Rabinowitz, F. Perbet, F. Song, C. Zhang, S. M. A. Eslami, and M. Botvinick, "Machine theory of mind," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR, 10–15 Jul 2018, pp. 4218–4227. [Online]. Available: http://proceedings.mlr.press/v80/rabinowitz18a.html

[78] D. Hughes, A. Agarwal, Y. Guo1, and K. Sycara1, "Inferring non-stationary human preferences for human-agent teams," 2020.

[79] S. Barrett, A. Rosenfeld, S. Kraus, and P. Stone, "Making friends on the fly: Cooperating with new teammates," *Artificial Intelligence*, October 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370216301266

[80] A. I. Goldman *et al.*, "Theory of mind."

[81] C. L. Baker, R. Saxe, and J. B. Tenenbaum, "Action understanding as inverse planning," *Cognition*, vol. 113, no. 3, pp. 329–349, 2009.

[82] T. Shu and Y. Tian, "M3rl: Mind-aware multi-agent management reinforcement learning," in *7th International Conference on Learning Representations (ICLR)*, 2019.

[83] J. Jara-Ettinger, "Theory of mind as inverse reinforcement learning," *Current Opinion in Behavioral Sciences*, vol. 29, pp. 105–110, 2019.

[84] J. Mendez, S. Shivkumar, and E. Eaton, "Lifelong inverse reinforcement learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 4502–4513.

[85] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning."

[86] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 1.

[87] A. Y. Ng, S. J. Russell, *et al.*, "Algorithms for inverse reinforcement learning."

[88] G. Neu and C. Szepesvári, "Apprenticeship learning using inverse reinforcement learning and gradient methods," *arXiv preprint arXiv:1206.5264*, 2012.

[89] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." 2008.

[90] G. Papamakarios, T. Pavlakou, and I. Murray, "Masked autoregressive flow for density estimation," in *Advances in Neural Information Processing Systems*, 2017, pp. 2338–2347.

[91] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," *arXiv preprint arXiv:1912.02762*, 2019.

[92] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[93] D. J. Rezende and S. Mohamed, “Variational inference with normalizing flows,” *arXiv preprint arXiv:1505.05770*, 2015.
[94] O. Rippel and R. P. Adams, “High-dimensional probability estimation with deep density models,” *arXiv preprint arXiv:1302.5125*, 2013.
[95] L. Dinh, J. Sohl-Dickstein, and S. Bengio, “Density estimation using real nvp,” *arXiv preprint arXiv:1605.08803*, 2016.
[96] N. R. Bramley, P. Dayan, T. L. Griffiths, and D. A. Lagnado, “Formalizing neurath’s ship: Approximate algorithms for online causal learning.” *Psychological review*, vol. 124, no. 3, p. 301, 2017.
[97] M. Wen, J. G. Kuba, R. Lin, W. Zhang, Y. Wen, J. Wang, and Y. Yang, “Multi-agent reinforcement learning is a sequence modeling problem,” 2022. [Online]. Available: https://arxiv.org/abs/2205.14953
[98] E. Cunningham, R. Zabounidis, A. Agrawal, I. Fiterau, and D. Sheldon, “Normalizing flows across dimensions,” *arXiv preprint arXiv:2006.13070*, 2020.
[99] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
[100] A. Mahinpei, J. Clark, I. Lage, F. Doshi-Velez, and W. Pan, “Promises and pitfalls of black-box concept learning models,” *arXiv preprint arXiv:2106.13314*, 2021.
[101] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, “Class-balanced loss based on effective number of samples,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9268–9277.
[102] N. C. Wagener, B. Boots, and C.-A. Cheng, “Safe reinforcement learning using advantage-based intervention,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 630–10 640.
[103] J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S. Srinivasa, “Expert Intervention Learning,” *Autonomous Robots*, vol. 46, no. 1, pp. 99–113, Jan. 2022. [Online]. Available: https://doi.org/10.1007/s10514-021-10006-9
[104] F. Wang, B. Zhou, K. Chen, T. Fan, X. Zhang, J. Li, H. Tian, and J. Pan, “Intervention aided reinforcement learning for safe and practical policy optimization in navigation,” in *Conference on Robot Learning*. PMLR, 2018, pp. 410–421.
[105] O. Bastani, Y. Pu, and A. Solar-Lezama, “Verifiable reinforcement learning via policy extraction,” *Advances in neural information processing systems*, vol. 31, 2018.
[106] S. Greydanus, A. Koul, J. Dodge, and A. Fern, “Visualizing and understanding atari agents,” in *International conference on machine learning*. PMLR, 2018, pp. 1792–1801.

APPENDIX

Range: The range concept indicates whether an agent is within range of another agent *and* facing it, where the range is within 0.8 map units and the angle is within $\frac{\pi}{5}$ radians. This value is a one-hot encoded value (within range or not within range) for each opposing agent, such that the total number of output nodes for this concept is $2n$ where n is the number of agents on the opposing team.

Strategy: The strategy concept indicates what team-level strategy the attacking team is following: *left*, *right*, or *random*. The strategy refers to the actions that an attacking team will take; a left strategy indicates that agents will follow a trajectory along the left side of the map leading to the objective, a right strategy indicates that agents will follow a trajectory along the right side of the map, and random indicates that agents will execute random actions. This concept is one-hot encoded and requires p output nodes, where p is the number of strategies – 3 in this work.

Target: The target concept indicates which agent on the opposing team an agent is currently *targeting*. The goal of this concept is to overcome the issue of oscillating targets, e.g., if an agent is equally close to multiple agents this can lead to oscillating behavior where the ego agent is unsure which other agent to pursue, and flips between them as the distance changes. During training, the targeted agent is initially selected to be the closest opposing agent and is only updated when that agent has been tagged. As with the *range* concept, this is a one-hot encoded value with $2n$ output nodes.

Orientation: The orientation concept is a continuous value representing the relative angle offset between an agent to each other agent on the opposing team. This value consists of n nodes where n is the number of agents on the opposing team.

Position: The position concept is a continuous value representing the relative Euclidean distance between an agent and each other agent on the opposing team. This value consists of n nodes where n is the number of agents on the opposing team.

Models are trained using a centralized-training-decentralized-execution approach, where a single policy is trained for all agents, and then executed individually for each agent during rollouts. True concept values are provided during training via an oracle function $V(\cdot)$ and used to compute the corresponding auxiliary loss, as well as the true concept values for intervention during intervened evaluation rollouts. Attacker strategies during training are sampled with equal probability from the set $\{$*left*, *right*, *random*$\}$, with sampled Left and Right strategies shown in Fig. 7. The *random* strategy is utilized to encourage the defenders to develop strategies in which they are free to pursue individual attackers, as opposed to remaining stationary near the objective. All policies were trained for 10M timesteps, after which the best policy checkpoint was taken – necessary since some models experienced forgetting and instability. Extensive hyperparameter optimization was performed, with the selected hyperparamters shown in Table V.

Table III is an expanded table showing the concept errors for each model in each scenario type with the addition of the standard error, indicating fairly consistent predictions. Table IV shows the win rate and intervened win rates for each model in each scenario type, with the addition of showing statistical significance. These win rates are computed over 100 evaluation episodes in simulation, and 20 episode evaluations in the real-world. The bolded value represents which model achieves the highest significant win rate for a given scenario type, with significance determined by the Fisher exact test and a $p < 0.05$. In the case of interventions, the only model which exhibits statistically significant improvement to the win rate after intervention is the 2v2 hard concept policy model, increasing from 25% to 95% – largely due to the small number of evaluation policy rollouts (20). We note that while the soft concept policy's intervened win rate may not be a statistically significant decrease, it is an interesting observation that interventions do not increase the win rate. We conjecture that this is because the residual layer encodes observation information which is affected by the distributional shift from simulation to the real-world, and despite being able to intervene over a subset of the concepts that are encoded, this is not enough to overcome the distribution gap. Furthermore, Table IV indicates that the intervened hard concept policy model yields an improved win rate in the real world than the simulation. This is an interesting outcome, and we conjecture that this is due to the real-world dynamics making it easier for the defenders to win assuming correct concept predictions. For example, the attackers can move and turn faster in simulation as they are not subject to real-world physics, and this makes it harder for the defenders to tag them.

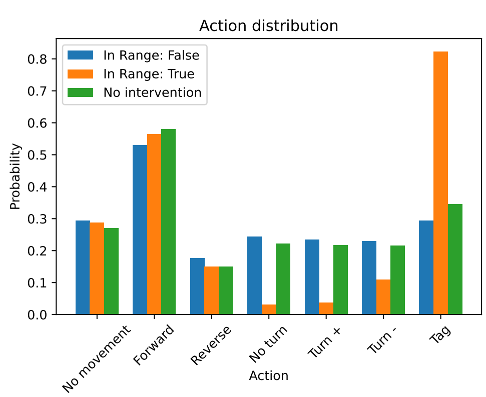


Fig. 4: Action distribution when intervening on the *Range* concept. Interventions are performed over the 3v3 hard concept policy model in simulation over 100 episodes. The *Range* concept is intervened such that the resulting value is always True or False, regardless of what the correct value is.

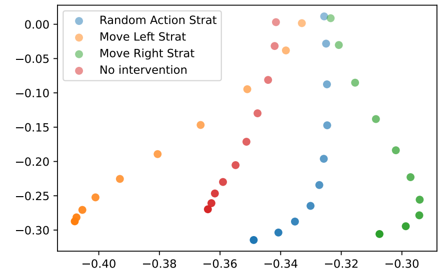


Fig. 5: Intervening on the *Strategy* concept for the 3v3 hard concept policy model in simulation. Each point is the average position of all the defenders as an episode progresses, where lighter values indicate earlier in the episode, averaged over 100 episodes.

Figure 4 shows the distribution of concept policy actions when concepts are intervened so as to always predict the same value, regardless of being correct or not. The goal of such an experiment is to produce a meaningful difference in the distribution of actions produced by the policy; if the defenders always predict False for the *Range* concept for example, we would expect that they would perform the tag action fewer times than if *Range* was always True. As seen in Fig. 4, this appears to hold and we can see that the frequency of the tag action is noticeably higher when *Range* is set to always be True. We conjecture that the influence of *Range* being True is much higher than False due to the concept itself being imbalanced – it is False far more often than True. As a result, the concept being True is more informative and can be directly associated to the act of tagging more readily than False.

Similarly, Fig. 5 shows the average position of all defenders during the course of an episode when the *Strategy* concept is intervened so as to take different values. When *Strategy* is set to always predict Left, we can see this yields an average position left of the No Intervention case, while Right yields an average position to the right of No Intervention. Intuitively, this shows that the defending agents act in response to the predicted *Strategy*, i.e., if the attackers are following a left strategy then the defenders move to the left to intercept.

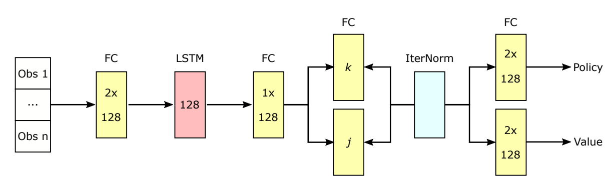


Fig. 6: The common model architecture shared by all models, which vary only in j and k.

The common model architecture shared by all models is shown in Fig. 6, varying only in the size of the concept and residual layers (as described in the main paper). The observations of each agent in the opposing team are stacked and fed through a series of FC layers. This is then passed through a recurrent LSTM layer to capture temporal information, and

	Setup	Model	*Range*	*Strategy*	*Target*	*Orientation*	*Position*
Simulation	2v2	Soft	0.03 ± 0.0032	0.04 ± 0.0060	0.24 ± 0.012		
		Hard	0.04 ± 0.0040	0.07 ± 0.0066	0.2 ± 0.012	0.10 ± 0.0071	0.11 ± 0.014
		Base					
	3v3	Soft	0.03 ± 0.0029	0.10 ± 0.0085	0.17 ± 0.0109		
		Hard	0.03 ± 0.0031	0.13 ± 0.098	0.23 ± 0.012	0.11 ± 0.0071	0.14 ± 0.0014
		Base					
	5v5	Soft	0.02 ± 0.0022	0.25 ± 0.012	0.48 ± 0.013		
		Hard	0.03 ± 0.0021	0.14 ± 0.012	0.13 ± 0.0097	0.11 ± 0.0063	0.21 ± 0.0142
		Base					
Real	2v2	Soft	0.02 ± 0.0068	0.88 ± 0.010	0.02 ± 0.021		
		Hard	0.04 ± 0.0074	0.81 ± 0.025	0.01 ± 0.0059	3.33 ± 0.21	0.08 ± 0.0091
		Base					

TABLE III: The concept errors (mean and standard error) for our proposed models (Soft and Hard) and a baseline without concepts (Base). The Hard model is trained over all concepts, the Soft model over a subset, and the Base model with none. *Range*, *Strategy*, and *Target* are discrete concepts and as such the error shown is the error in accuracy score, while *Orientation* and *Position* are continuous and indicate mean squared error. *Orientation* is in radians and *Position* is a unit-less value in $[-1, 1]$. Errors are computed over 100 episode rollouts for each model for simulated, and 20 rollouts for real-world.

	Setup	Model	WR	Intervened WR
Simulation	2v2	Soft	51%	55%
		Hard	**83%**	84%
		Base	34%	
	3v3	Soft	55%	57%
		Hard	**74%**	80%
		Base	16%	
	5v5	Soft	32%	40%
		Hard	**78%**	86%
		Base	31%	
Real	2v2	Soft	10%	0%
		Hard	25%	95%
		Base	35%	

TABLE IV: The win rate and intervened win rate for each model in each scenario type. The best performing model for each scenario type (according to Fisher exact with $p < 0.05$) is highlighted in bold.

then split into the concept and residual layers. Whitening is performed via iterative normalization over the concatenated concept and residual layers, which are then passed into the policy and value heads consisting of more fully connected layers. Each group of (2x 128) fully connected layers are followed by a ReLU activation, with the group-wise softmax following the concept layer after the IterNorm for discrete concepts only. The auxiliary loss $L^c(\theta)$ is computed over the concepts after the IterNorm layer.

Learning Rate Schedule	1e-3 at t=0 to 1e-4 at t=10M
Entropy Schedule	0.1 at t=0 to 0.01 at t=10M
LSTM Max Sequence Length	50
Batch Size	10240
SGD Minibatch Size	1600 (Seq. Length × 32)
Concept Loss Coeff.	10
T	2
Optimizer	Adam
β_1, β_2	0.9, 0.999

TABLE V: Hyperparameters used for each model type.

The hyperparameters used for each model during training are given in Table V. The learning rate and entropy values used a linear scheduler where the values were decreased as training progressed.

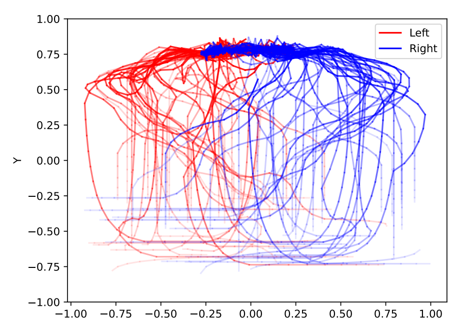


Fig. 7: Sampled attacker strategies from the Left and Right strategies. Lighter points indicate agent positions earlier in the episode. The objective is at the top of the screen. Attackers randomly spawn in the lower half of the environment.

Hearing Touch: Using Contact Microphones for Robot Manipulation

Shaden Alshammari[1], Victoria Dean[2], Tess Hellebrekers[3], Pedro Morgado[4], Abhinav Gupta[2]

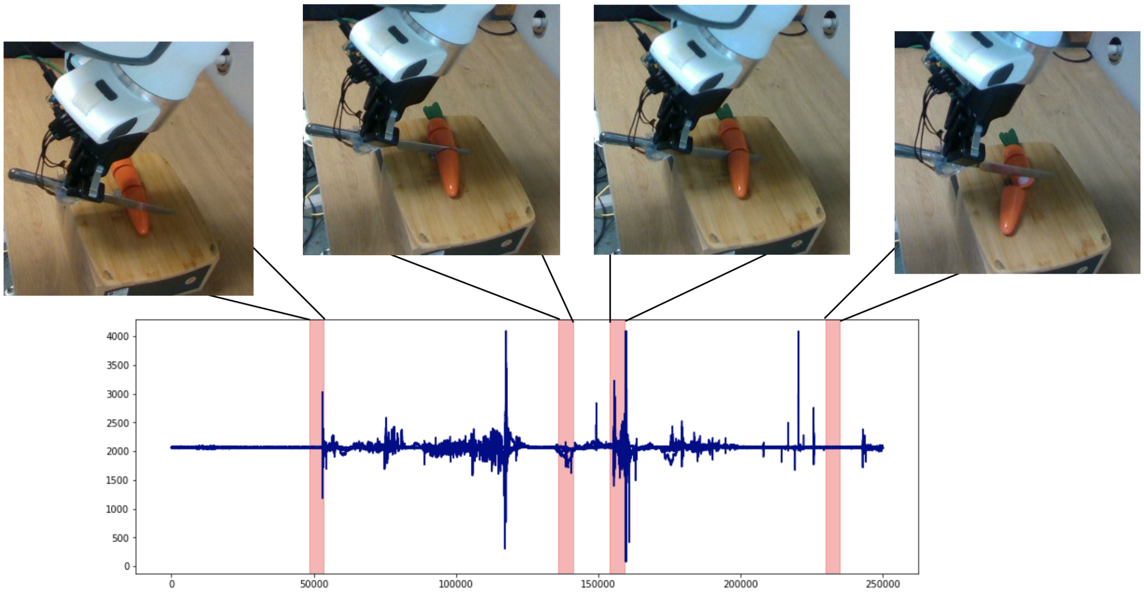


Figure: example for the audio recording during a chopping task

***Abstract*— Humans use all of their senses to comprehend their immediate physical environment, including how sound and action interact. However, microphones have not been widely used as a tactile sensor in robotics, whereas visual and other sensors have. Using contact microphones, which record high frequency vibrations by coming into contact with solid objects, we investigate contact audio as an alternative tactile modality. Because it can cross domains more effectively than vision, audio modality is more robust. Additionally, it can accurately capture interactions with various objects and materials. We investigate the use of contact microphones as a sensor in tasks centered around the kitchen.**

***Index Terms*— tactile manipulation, audio in robotics, task classification.**

I. INTRODUCTION

Humans manipulate objects using all of their senses, including sound and touch: audio can indicate whether or not the door has been unlocked or an egg has been properly cracked. Prior work has shown that humans can use auditory feedback alone to categorize types of events and infer continuous aspects of these events, such as the length of a wooden dowel being struck [1]. However, microphones remain underexplored in robotics, especially their potential as tactile vibration sensors.

In this work, we investigate contact audio as an alternative tactile modality for complex manipulation tasks that are challenging from vision alone. Contact microphones record vibrations of anything in direct contact at a high-frequency (1000 times higher frequency than the next common tactile sensor [2]). This makes them well-suited to use as tactile sensors when interacting with objects in manipulation. Furthermore, contact audio is immune to many aspects of environment variation that vision is plagued by, such as lighting and color variation, making it promising for transfer learning and multi-task settings that are common in robotics.

We mount the contact microphones on a robot gripper and gather data on a chopping task (see Fig. III).Then, we extract a representation from both audio and video. We implement real-robot experiments using the k-nearest neighbors algorithm. We compare the performance of the robot with and without using audio.

[1] Massachusetts Institute of Technology
[2] Robotics Institute, Carnegie Mellon University
[3] Meta AI Research
[4] University of Wisconsin-Madison

II. RELATED WORK

A. Audio in computer vision

There is a strong correlation between objects, actions and sounds. For example, Owens et al. propose the task of predicting the sound an object makes when scratching various objects based on silent videos, and develop an algorithm that generates the audio [3]. Moreover, it is important to learn representations based on both video and audio for action classification. Morgado et al. develop a contrastive learning approach Audio-Visual Instance Discrimination with Cross-Modal Agreement (AVID) to learn audio-visual representations from video and audio [4].

Fig. 1. Contact microphones setup

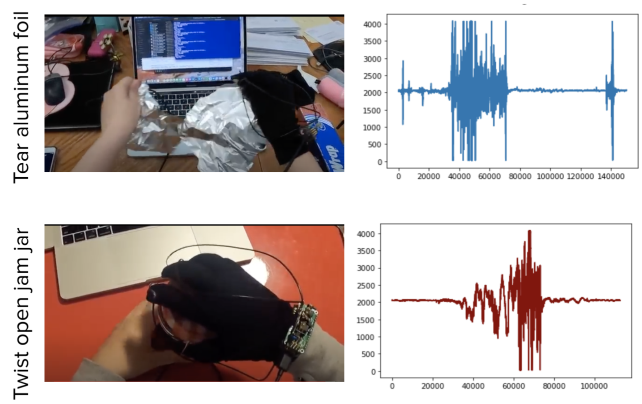


Fig. 2. task examples collected using the contact microphones on the glove

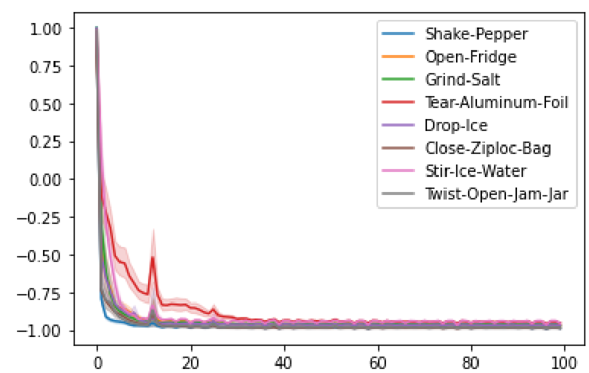


Fig. 3. average tasks signals after prepossessing with 95% confidence interval

task	number of examples	total time
Tear Aluminum Foil	118	78.17s
Stir Ice Water	79	50.99s
Open Fridge	59	38.52s
Shake Pepper	86	55.19s
Grind Salt	80	54.90s
Twist Open Jam Jar	46	29.93s
Close Ziploc	283	188.91s
Drop Ice	54	35.33s

III. CONTACT MICROPHONES AS TACTILE SENSORS

A contact microphone, or a piezo microphone records high frequency vibrations when coming into contact with solid objects. Contact microphones only capture sound that is carried by object in contact which doesn't include air vibrations unlike typical air microphones. They can be used for a variety of tasks, including amplifying the sound from acoustic musical instruments and triggering electronic samples. They can also record sound in difficult settings, such as underwater or under high pressure.

Contact microphones are feasible for robots as tactile sensors because they can precisely record interactions with different objects and materials. They also can effectively cross domains unlike vision, which depends on a variety of factors like lightning. Lastly, their price is extremely low (about $1 for a piece). For our tasks, we attach contact microphones to two types of objects: glove (see figure III) and a robot gripper.

IV. EXPERIMENTAL SETUP

A. Chopping task

We investigate the impact of using contact microphones in learning chopping tasks. We fix contact microphones on robot grippers and a knife on top of them. Then, while chopping a plastic carrot with two designated chopping areas, we collected audio, visual, and action data for the robot.

B. Dataset details

We collect a dataset using the glove microphone for the 8 tasks described below over different settings.

V. METHODS

We are interested in two types of tasks: classification and robot manipulation. For task classification, we have a dataset of more than 800 examples on a kitchen centric tasks (see table above). For task classification, we use two approaches: the first is shallow neural networks since our dataset is relatively small and the audio data is low dimensional after the prepossessing. The second approach is by fine-tuning a multimodal model on the AVID representation.

For robot manipulation, we collected 12 examples of a chopping task on a franka arm robot which has contact microphones attached to its gripper. The data consists of audio clips, video, and robot actions. We use the k-nearest neighbors (k-NN) regression for manipulation. We consider

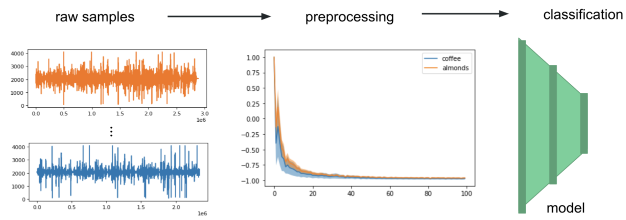


Fig. 4. task classification pipeline

L2 as a distance metric over multiple representations. The first representation is simply using FFT for 1-second clips and pooling 100 parts of it uniformly. The second representation is extracted from the trained classifier described above.

VI. Results

A. Classification

We classified the dataset collected using the glove as described in IV-B. The audio and video clips have been converted to 1-second clips. For the MLP model, the audio signal is prepossessed using Fast Fourier Transform (FFT). The table below shows the classification accuracy for different models

model	input modality	accuracy
MLP (3 layers)	audio only	63%
AVID fine-tuning	audio only	74%
AVID fine-tuning	video only	96%
AVID fine-tuning	audio & video	100%

B. Manipulation

We collect tele-operated demonstrations and create a dataset with the recorded contact audio and images. At test time, we use the k-nearest neighbors (KNN) algorithm to infer which part of the demonstrations most closely matches the current audio-visual context in an embedding space, and these nearest neighbors can be used to select actions. VI-B visualizes the nearest neighbors throughout a trial using two embedding spaces: audio-only with fast fourier transform (left) and a neural network trained on a classification task with audio-visual input. The figure shows that the nearest neighbors retrieved by FFT are more commonly in chopping regions (denoted by audio peaks) than their classifier embedding counterparts, perhaps due to information loss in the latter's embedding. In future work, we aim to extend these results to a broader set of policy learning methods, embedding spaces, and tasks.

VII. Conclusion

In this work, we studied contact audio as an alternate tactile modality for complicated manipulation tasks . For future work, an audio-visual representation for the tasks. We also want to experiment other manipulation learning techniques such as imitation learning.

Acknowledgements

Shaden would like to thank Victoria, Tess, Pedro and Abhinav for their guidance and support throughout the project.She would also like to thank Rachel Burcin and Dr. John Dolan for their continuous support and organization of this program. This work is supported in part by the KAUST Gifted Student's Program (KGSP).

References

[1] S. Clarke, T. Rhodes, C. G. Atkeson, and O. Kroemer, “Learning audio feedback for estimating amount and flow of granular material,” *Proceedings of Machine Learning Research*, vol. 87, 2018.

[2] R. Bhirangi, T. Hellebrekers, C. Majidi, and A. Gupta, “Reskin: versatile, replaceable, lasting tactile skins,” *arXiv preprint arXiv:2111.00071*, 2021.

[3] A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman, “Visually indicated sounds,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2405–2413.

[4] P. Morgado, N. Vasconcelos, and I. Misra, “Audio-visual instance discrimination with cross-modal agreement,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 475–12 486.

[5] V. Dean, S. Tulsiani, and A. Gupta, “See, hear, explore: Curiosity via audio-visual association,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 14 961–14 972, 2020.

[6] H. Zhao, C. Gan, W.-C. Ma, and A. Torralba, “The sound of motions,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1735–1744.

[7] D. Gandhi, A. Gupta, and L. Pinto, “Swoosh! rattle! thump!–actions that sound,” *arXiv preprint arXiv:2007.01851*, 2020.

[8] Sweetwater, “Contact microphone,” Apr 2017. [Online]. Available: https://www.sweetwater.com/insync/contact-microphone/

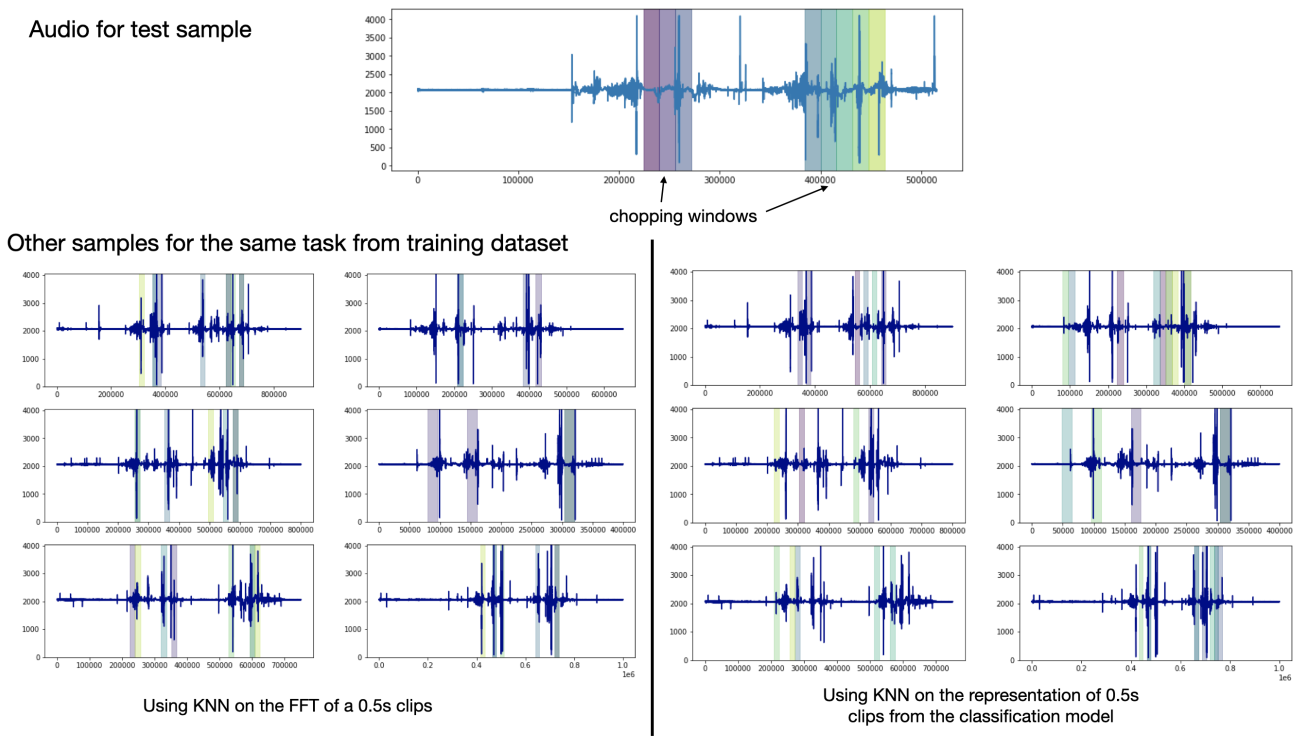


Fig. 5. KNN functions as a manipulation policy mapping the current audio-visual context to closest points from demonstrations. We color-code the test trial (top) and show each segment's three closest neighbors (bottom).

Active Probing and Influencing Human Behaviors Via Autonomous Agents

Shuangge Wang[1], Yiwei Lyu[2], John M. Dolan[3]

Abstract— **Autonomous agents (robots) face tremendous challenges while interacting with heterogeneous human agents in close proximity. One of these challenges is that the autonomous agent does not have an accurate model tailored to the specific human that the autonomous agent is interacting with, which could sometimes result in inefficient human-robot interaction and suboptimal system dynamics. Developing an online method to enable the autonomous agent to learn information about the human model is therefore an ongoing research goal. Existing approaches position the robot as a passive learner in the environment to observe the physical states and the associated human response. This passive design, however, only allows the robot to obtain information that the human chooses to exhibit, which sometimes doesn't capture the human's full intention. In this work, we present an online optimization-based probing procedure for the autonomous agent to clarify its belief about the human model in an active manner. By optimizing an information radius, the autonomous agent chooses the action that most challenges its current conviction. This procedure allows the autonomous agent to actively probe the human agents to reveal information that's previously unavailable to the autonomous agent. With this gathered information, the autonomous agent can interactively influence the human agent for some designated objectives. Our main contributions include a coherent theoretical framework that unifies the probing and influence procedures and two case studies in autonomous driving that show how active probing can help to create better participant experience during influence, like higher efficiency or less perturbations.**

I. Introduction

It is imperative for robots to behave reactively in a human-present environment because all safety specifications ought to be met. An autonomous vehicle, for instance, should yield to a human vehicle trying to nudge in front of it; a reconnaissance drone should avoid adversarial behaviors. Robots, however, are usually not designed to behave purely in a reactive manner because it makes them too conservative. Consider a scenario of autonomous driving (Fig. 1) where the human vehicle is traveling in the outer lane (lower), but at a fast enough speed that it's better, for efficiency purposes, to switch to the inner lane (upper). Many human drivers don't have the awareness to make this transition because they are usually egocentric, even subconsciously, in that they would rather remain in their current lane unless blocked by some other vehicles. Some works, therefore, have proposed to use autonomous vehicles to influence the fast human vehicles to drive in the inner lane by blocking them.

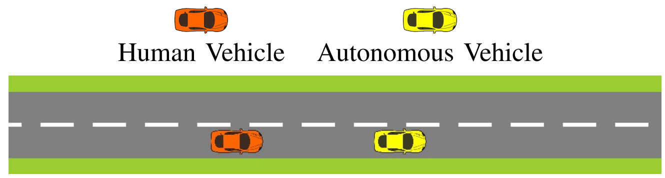


Fig. 1: Both vehicles currently travel to the right in the outer lane (lower). Autonomous vehicle (yellow) intends to influence human vehicle (orange) with intention to drive fast to inner lane (upper).

Since such influence is exerted in close proximity, the autonomous agent needs an accurate human model. Although techniques like Inverse Reinforcement Learning (IRL) can produce a generally reasonable model, this falls short when an autonomous agent is interacting with a human in close proximity because the model may not capture characteristics specific to the human agent that the robot is interacting with. For instance, in Fig. 1, the autonomous vehicle is interested in, rather precisely, the desired travel velocity of the human vehicle, and each human differs from another in their desired velocities.

Existing online approaches tackle this problem by positioning the autonomous agent as a passive observer, in which it observes the environmental states and their associated human response and then chooses the model that best explains this correlation. The issue with this design is that the autonomous agent is passive, so it only has access to information that the human agent chooses to exhibit. Hence, the autonomous agent can only make decisions based on the human information that's readily available. In Fig. 1 for instance, a passive autonomous vehicle would presume the human vehicle intends to travel at most as fast as itself, whereas in reality the human could want to drive faster, only to be blocked by the autonomous vehicle.

In this work, we enable autonomous agents to leverage their own actions to estimate the human internal model by actively interacting with the human agent to reveal more information. Rather than relying on passive observations, the autonomous agent can actually account for the fact that the human will react to its actions, so the autonomous agent can "probe", i.e., select the actions that will trigger human reactions that in turn will best challenge its initial belief. By probing iteratively, the autonomous agent converges to an increasingly accurate human model. Then, based on the probed information, the autonomous agent can actively influ-

[1]Shuangge Wang is with the Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA, 90089 USA. Email: larrywan@usc.edu

[2]Yiwei Lyu is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA. Email: yiweilyu@andrew.cmu.edu

[3]John M. Dolan is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA. Email: jdolan@andrew.cmu.edu

ence other agents for some designated objectives, like higher efficiency or better driving experience. Our key contributions in this work are: 1) a coherent theoretical framework that unifies the probing and influencing procedures; 2) a proven solvable trajectory-planning optimization with reasonably mild assumptions; 3) two case studies as application examples in the domain of autonomous driving with numerical simulations used to demonstrate the precision of probing results and efficacy in creating better participant driving experience during influence.

II. Related Work

Human behavior, in general, is determined by an internal model that the autonomous agent cannot directly access [1], [2]. Such an internal model might characterize human's intentions, preferences, objectives, strategies, etc. To exert the influence, an autonomous agent would have to interact with different human agents in close proximity, who are heterogeneous agents that differ significantly in their internal models. Works in robotics and perception have focused on estimating these internal models using algorithms based on observations of human's actions, such as intent-driven behavior prediction [3]–[9], IRL [10]–[14], hidden model prediction [15], affective state estimation [16], and activity recognition [17]. Although the human model derived from the above methods performs generally reasonably, it might not capture specific characteristics of the human agent that the autonomous agent is interacting with. The autonomous agent, therefore, needs an online procedure to learn the model specially tailored to the human agent that the autonomous agent is interacting with.

Some online approaches frame this problem as a Partially Observable Markov Decision Process (POMDP) [18]–[20], in which the autonomous agent represents the human's intent through a parametric model, inferred through Markovian or Bayesian estimation of the hidden parameters of the internal models from observations of the physical states of the world [21]–[24]. In this paradigm, the autonomous agent is mainly a reactive agent present in the environment to merely observe, which hugely sacrifices the the robot's personal agency to initiate action to actively reveal information about the human.

Some existing works enable active probing for interactive motion planning by incorporating a heuristic active information gathering objective, e.g., information entropy, into the autonomous agent's trajectory optimization framework for human value function parameter estimation [25], [26]. Building upon this work, we allow the autonomous agent to optimize the information radius, i.e., the cohesion between two beliefs, relative to its latest belief of the human model, so instead of having a fixed reference belief like in [26] the autonomous agent aims to maximize the information radius relative to a dynamic reference, its current belief, at every time iteration.

III. Theory

A. Human-Robot Joint Dynamics

For all notations below, we use subscripts to denote the time step and superscripts to capture the attributes' ownership (human or robot). In a human-robot joint system, we define the state vector as $s_t \in \mathbb{R}^n$, the robot's input vector as $u_t^{\mathcal{R}} \in \mathbb{U}^{\mathcal{R}} \subseteq \mathbb{R}^{m^{\mathcal{R}}}$, confined to admissible control space $\mathbb{U}^{\mathcal{R}}$, the human's input vector as $u_t^{\mathcal{H}} \in \mathbb{U}^{\mathcal{H}} \subseteq \mathbb{R}^{m^{\mathcal{H}}}$, confined to admissible control space $\mathbb{U}^{\mathcal{H}}$, and finally the discrete-time control-affine dynamics of the joint system as

$$s_{t+1} = f(s_t) + M^{\mathcal{R}}(s_t)u_t^{\mathcal{R}} + M^{\mathcal{H}}(s_t)u_t^{\mathcal{H}} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ captures the non-linear autonomous dynamics and $M^{\mathcal{R}} : \mathbb{R}^n \to \mathbb{R}^{n \times m^{\mathcal{R}}}$ and $M^{\mathcal{H}} : \mathbb{R}^n \to \mathbb{R}^{n \times m^{\mathcal{H}}}$ are state-dependent input transformation matrices for robot and human respectively [25].

B. Belief Update

The autonomous agent possesses a belief of φ that characterizes the human agent's utility function $r_\varphi^{\mathcal{H}} : \mathbb{R}^n \to \mathbb{R}$. For driving scenarios, a typical φ could characterize the desired velocity of the human vehicle, and a typical $r_\varphi^{\mathcal{H}}$ would include features like safety and speed. We generalize the autonomous agent's belief by proposing a non-parametric representation which can approximate a wider range of distributions. At time t, belief of φ is defined as bel_t with finite domain space Φ. The autonomous agent updates this belief via a particle-filtering recursion [27]

$$bel_{t+1}(\varphi) \propto bel_t(\varphi) \cdot p(u_t^{\mathcal{H}} | s_t, u_t^{\mathcal{R}}, \varphi), \; \forall \varphi \in \Phi \tag{2}$$

where the conditional probability is obtained through a softmax operation based on the Boltzmann model of exponential likeliness of human actions with greater utility [9], [13]

$$p(u_t^{\mathcal{H}} | s_t, u_t^{\mathcal{R}}, \varphi) = \frac{e^{r_\varphi^{\mathcal{H}}\left(f(s_t) + M^{\mathcal{R}}(s_t)u_t^{\mathcal{R}} + M^{\mathcal{H}}(s_t)u_t^{\mathcal{H}}\right)}}{\sum_{\tilde{u}_t^{\mathcal{H}} \in \mathbb{U}^{\mathcal{H}}} e^{r_\varphi^{\mathcal{H}}\left(f(s_t) + M^{\mathcal{R}}(s_t)u_t^{\mathcal{R}} + M^{\mathcal{H}}(s_t)\tilde{u}_t^{\mathcal{H}}\right)}} \tag{3}$$

in which $\mathbb{U}^{\mathcal{H}}$ is discretized for softmax normalization. The complete belief update algorithm is shown in algorithm 1.

Algorithm 1 Belief Update

Input: $bel_t, s_t, u_t^{\mathcal{R}}, u_t^{\mathcal{H}}$

$\eta \leftarrow 0$
for all $\varphi \in \Phi$ **do**
  $r \leftarrow e^{r_\varphi^{\mathcal{H}}\left(f(s_t) + M^{\mathcal{R}}(s_t)u_t^{\mathcal{R}} + M^{\mathcal{H}}(s_t)u_t^{\mathcal{H}}\right)}$ {Boltzmann}
  $\tilde{r} \leftarrow \sum_{\tilde{u}_t^{\mathcal{H}} \in \mathbb{U}^{\mathcal{H}}} e^{r_\varphi^{\mathcal{H}}\left(f(s_t) + M^{\mathcal{R}}(s_t)u_t^{\mathcal{R}} + M^{\mathcal{H}}(s_t)\tilde{u}_t^{\mathcal{H}}\right)}$
  $bel_{t+1}(\varphi) \leftarrow bel_t(\varphi) \cdot \frac{r}{\tilde{r}}$ {belief update}
  $\eta \leftarrow \eta + bel_{t+1}(\varphi)$
end for
for all $\varphi \in \Phi$ **do**
  $bel_{t+1}(\varphi) \leftarrow \frac{bel_{t+1}(\varphi)}{\eta}$ {belief normalization}
end for
return bel^{t+1}

C. Probing

The motivation behind probing is to allow the autonomous agent to actively interact with the human agent to reveal more information that was previously unavailable, meaning that the autonomous agent should choose actions that best challenge its current belief at every time step. Quantitatively, the autonomous agent chooses actions that maximize the information radius between its current belief and the projected belief if such actions are to be executed.

We introduce the Jensen-Shannon divergence (JSD) as a measure of information radius to quantify the cohesion between two beliefs, bel_a and bel_b [28], [29]

$$D_{\mathrm{JS}}[bel_a, bel_b] = \frac{D_{\mathrm{KL}}\left[bel_a : \overline{bel}_{a,b}\right] + D_{\mathrm{KL}}\left[bel_b : \overline{bel}_{a,b}\right]}{2} \tag{4}$$

where D_{KL} denotes the Kullback–Leibler divergence (KLD) [30], [31] and $\overline{bel}_{a,b}$ is the arithmetic mixture of bel_a and bel_b

$$D_{\mathrm{KL}}[bel_a : \overline{bel}_{a,b}] = \mathop{\mathbb{E}}_{\varphi \sim bel_a} \log\left(\frac{2 \cdot bel_a(\varphi)}{bel_a(\varphi) + bel_b(\varphi)}\right) \tag{5}$$

At state s_t, the autonomous agent predicts how the human agent, characterized by φ, will react to its action $u_t^{\mathcal{R}}$ using

$$Q(s_t, u_t^{\mathcal{R}}, \varphi) = \mathop{\arg\max}_{\tilde{u}_t^{\mathcal{H}} \in \mathbb{U}^{\mathcal{R}}} r_\varphi^{\mathcal{H}}(f(s_t) + M^{\mathcal{R}}(s_t) u_t^{\mathcal{R}} + M^{\mathcal{H}}(s_t) \tilde{u}_t^{\mathcal{H}}) \tag{6}$$

We solve the probing problem using Model Predictive Control (MPC) with finite horizon T, in which the autonomous agent chooses a sequence of actions that optimizes the JSD between the current belief and the projected belief on the horizon

$$\max_{u_{0:T-1}^{\mathcal{R}}} \mathop{\mathbb{E}}_{\varphi \sim bel_0} \sum_{t=0}^{T-1} D_{\mathrm{JS}}[bel_0, bel_{t+1}] - D_{\mathrm{JS}}[bel_0, bel_t] \tag{7a}$$

$$\text{s.t.} \quad s_0 = s_t, bel_0 = bel_t \tag{7b}$$

$$u_t^{\mathcal{H}} = Q(s_t, u_t^{\mathcal{R}}, \varphi) \tag{7c}$$

$$s_{t+1} = f(s_t) + M^{\mathcal{R}}(s_t) u_t^{\mathcal{R}} + M^{\mathcal{H}}(s_t) u_t^{\mathcal{H}} \tag{7d}$$

$$bel_{t+1}(\varphi) \propto bel_t(\varphi) \cdot p(u_t^{\mathcal{H}} | s_t, u_t^{\mathcal{R}}, \varphi) \tag{7e}$$

To ensure solvability, we prove that D_{JS}, which maps to $[0, \infty)$ in theory, is upper bounded in optimization (7).

Proof. **Boundedness**:

We first make a slight assumption that bel_0 is bounded and has compact support, hence

$$\sup_{\varphi \in \Phi} bel_0(\varphi) < \infty \wedge \inf_{\varphi \in \Phi} bel_0(\varphi) > 0 \tag{8}$$

which helps to substantiate the boundedness of KLD [32]. We will initialize the belief such that condition (8) is satisfied in section IV.

For induction hypothesis, we assume that $\forall a \in \{0, \ldots, T-1\}$, $\sup_{\varphi \in \Phi} bel_a(\varphi) < \infty$. Since $p(u_t^{\mathcal{H}} | s_t, u_t^{\mathcal{R}}, \varphi)$ maps to an image of $(0, 1)$, using condition (8) as base case, we have

$$\sup_{\varphi \in \Phi} bel_a(\varphi) < 1 < \infty, \ \forall a \in \{0, \ldots, T\} \tag{9}$$

By similar induction technique, we have

$$\inf_{\varphi \in \Phi} bel_a(\varphi) > 0, \ \forall a \in \{0, \ldots, T\} \tag{10}$$

Hence, we have extended condition (8) to

$$\sup_{\varphi \in \Phi} bel_a(\varphi) < \infty \wedge \inf_{\varphi \in \Phi} bel_a(\varphi) > 0, \ \forall a \in \{0, \ldots, T\} \tag{11}$$

Therefore, $\forall a \in \{0, \ldots, T\}$, $\exists \bar{s} = \sup_{\varphi \in \Phi} bel_a(\varphi)$ such that $0 < \bar{s} < \infty$. Similarly, $\forall a, b \in \{0, \ldots, T\}$, $\exists \underline{i} = \inf_{\varphi \in \Phi} bel_a(\varphi) + bel_b(\varphi)$ such that $0 < \underline{i} < \infty$.

Therefore, by equation (5), we have $\forall a, b \in \{0, \ldots, T\}$

$$\begin{aligned} D_{\mathrm{KL}}[bel_a : \overline{bel}_{a,b}] &= \mathop{\mathbb{E}}_{\varphi \sim bel_a} \log\left(\frac{2 \cdot bel_a(\varphi)}{bel_a(\varphi) + bel_b(\varphi)}\right) \\ &\leq \mathop{\mathbb{E}}_{\varphi \sim bel_a} \sup_{\varphi \in \Phi} \log\left(\frac{2 \cdot bel_a(\varphi)}{bel_a(\varphi) + bel_b(\varphi)}\right) \\ &\leq \log(2 \cdot \bar{s}) - \log(\underline{i}) < \infty \end{aligned} \tag{12}$$

By symmetry, $D_{\mathrm{KL}}[bel_b : \overline{bel}_{a,b}] < \infty$ can be easily proved using the same technique, which together concludes the boundedness of JSD. □

We propose a dynamic-programming-based approach to optimize equation (7). Although the computational complexity grows exponentially with respect to the state dimension, we argue that successfully reasoning about human-robot interactions over a short horizon does not require a full-fidelity model of the joint dynamics, so highly informative insights can still be obtained tractably via approximation. We define the value function of executing n consecutive controls starting from time k as

$$V(k, n) = \mathop{\mathbb{E}}_{\varphi \sim bel_0} \sum_{t=k}^{k+n-1} D_{\mathrm{JS}}[bel_0, bel_{t+1}] - D_{\mathrm{JS}}[bel_0, bel_t] \tag{13}$$

The value function on the horizon therefore satisfies

$$\begin{aligned} V(0, T) &= \mathop{\mathbb{E}}_{\varphi \sim bel_0} \sum_{t=0}^{k-1} D_{\mathrm{JS}}[bel_0, bel_{t+1}] - D_{\mathrm{JS}}[bel_0, bel_t] \\ &+ \mathop{\mathbb{E}}_{\varphi \sim bel_0} \sum_{t=k}^{T-1} D_{\mathrm{JS}}[bel_0, bel_{t+1}] - D_{\mathrm{JS}}[bel_0, bel_t] \\ &= V(0, k) + V(k, T-k), \ \forall k \in \{0, \ldots, T\} \end{aligned} \tag{14}$$

which shows that the path-dependency fits a Bellman optimality equation [33].

Therefore, an optimal value function and control policy can be obtained in polynomial time by backtracking the Hamilton–Jacobi–Bellman (HJB) equation [34]

$$V(t, T-t) = \max_{u_t^{\mathcal{R}} \in \mathbb{U}^{\mathcal{R}}} \left\{ V(t, 1) + V(t+1, T-t-1) \right\} \tag{15}$$

Following this policy, the autonomous agent interactively probes the human agent and gradually converges its belief until the change of JSD is too small. The autonomous agent then chooses $\hat{\varphi}$, which could be a linear combination of all $\varphi \in \Phi$ weighted by their $bel(\varphi)$ or simply the most likely $\varphi \in \Phi$, as the human model parameter.

D. Influence

We characterize an influence as a sequence of atomic objectives, each with a utility function, that accounts for a major influence if all executed in order, and we delegate the responsibility of planning these atomic objectives to some high-level planner. For each objective, we incorporate $\hat{\varphi}$ into the utility function for both robot and human.

$$\max_{u^{\mathcal{R}}_{0:T-1}} \sum_{t=0}^{T-1} r^{\mathcal{R}}_{\hat{\varphi}(s_{t+1})} \tag{16a}$$

$$\text{s.t.} \quad s_0 = s_t \tag{16b}$$

$$u^{\mathcal{H}}_t = Q(s_t, u^{\mathcal{R}}_t, \hat{\varphi}) \tag{16c}$$

$$s_{t+1} = f(s_t) + M^{\mathcal{R}}(s_t)u^{\mathcal{R}}_t + M^{\mathcal{H}}(s_t)u^{\mathcal{H}}_t \tag{16d}$$

Similarly, this optimization problem can be solved using HJB recursion in polynomial time.

IV. Simulation

In this section, we present two car-following-based scenarios in which probing and influencing can be used to facilitate better participant experience and optimality for human drivers. Both scenarios start with the human vehicle following the autonomous vehicle.

A. Ground Truth

To generate the ground truth trajectories for the human-driven vehicle, we use the intelligent driver model (IDM) [35]–[37], which is known to accurately imitate human driving behaviors.

$$u^{\mathcal{H}} = u_{\max}\left[1 - \left(\frac{v^{\mathcal{H}}}{v_{\text{des}}}\right)^4 - \left(\frac{d_{\text{des}}}{x^{\mathcal{R}} - x^{\mathcal{H}}}\right)^2\right] \tag{17}$$

in which

$$d_{\text{des}} = d_{\min} + \tau_{\text{gap}} \cdot v^{\mathcal{H}} - \frac{v^{\mathcal{H}} \cdot (v^{\mathcal{H}} - v^{\mathcal{R}})}{2\sqrt{a_{\max} \cdot b_{\text{pref}}}} \tag{18}$$

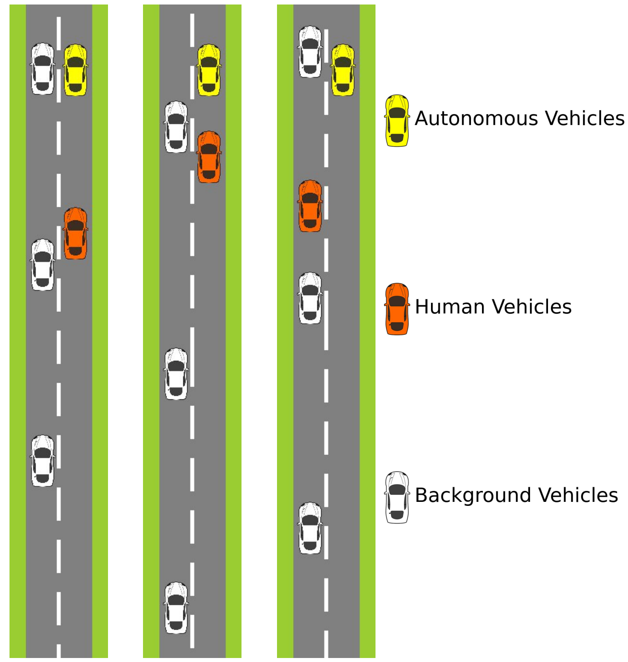
where superscripted notations are system dynamics and subscripted notations are constant parameters. We assume that the vehicles will maintain their driving style, so the constant parameters above are static over time. Without loss of generality, we also use IDM to model other background vehicles in the environment.

B. Exploitation and Exploration

To balance exploitation and exploration, the autonomous vehicle alternates between 5 s of passive observation and 5 s of active probing. We also set the MPC horizon to 5 s. Thanks to the boundedness of JSD, we can add a safety objective, $\lambda \cdot r^{\mathcal{R}}_{safe}(s_{t+1})$, on the autonomous agent's optimization to enforce some safety features, and we choose λ empirically.

C. Human Model

The autonomous vehicle models the human underlying utility using a combination of features, namely desired headway and desired velocity. For each scenario, we choose $|\Phi| = 30$ such that each $\varphi \in \Phi$ maps to a distinct desired velocity or desired headway, and we initialize them to a uniform distribution, which satisfies condition (8).



(a) Phase 1 (b) Phase 2 (c) Phase 3

Fig. 2: Phase 1: Autonomous vehicle maintains velocity. Phase 2: Autonomous vehicle decreases velocity to block the human vehicle. Phase 3: Human vehicle merges due to blocking. All vehicles are traveling upwards.

D. Scenario 1: Influence fast drivers to switch lane

Consider a two-lane highway (Fig. 2a) with an inner lane (left) and an outer lane (right). Here, we cause the autonomous vehicle to actively probe the desired velocity of the human vehicle. If the human vehicle exhibits the intention to travel at a high velocity, the autonomous vehicle will perform a series of maneuvers to help the human vehicle merge to the inner lane in the widest gap between the background vehicles. While approaching the widest gap, the autonomous vehicle slows down to block the human vehicle (Fig. 2b), and the human vehicle switches lanes shortly after that (Fig. 2c).

We choose the IDM parameters as $u_{\max} = 0.73\,\text{m/s}^2$, $b_{\text{pref}} = 1.67\,\text{m/s}^2$, $v_{\text{des}} = 25\,\text{m/s}$, $\tau_{\text{gap}} = 1.5\,\text{s}$, and $d_{\min} = 2\,\text{m}$. We start the car-following scenario with relative headway of $100\,\text{m}$, the autonomous vehicle and the human vehicle both traveling at $20\,\text{m/s}$. We also included a passive observing approach to compare as a baseline. Fig. 3 is a snapshot of the belief from two approaches taken every $10\,\text{s}$.

By $50\,\text{s}$, the active approach's peak happens at φ_{19}, which maps to a desired velocity of $23.56\,\text{m/s}$, which is close to the IDM parameter, v_{des}, of $25\,\text{m/s}$. In comparison, the passive observation baseline peaks at φ_{16} that maps to $19.86\,\text{m/s}$, which is very far from the ground truth. This is because the passive approach suffers from no exploration to trigger human reaction, so the autonomous vehicle will assume the human vehicle intends to travel only as fast as itself.

Leveraging this probed information, the autonomous vehi-

< TABLE OF CONTENTS

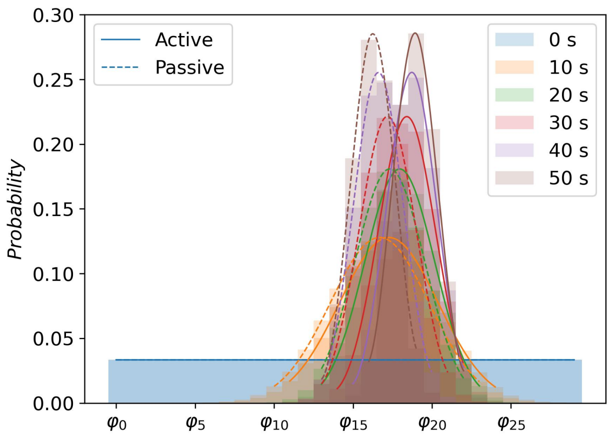


Fig. 3: Belief Snapshot

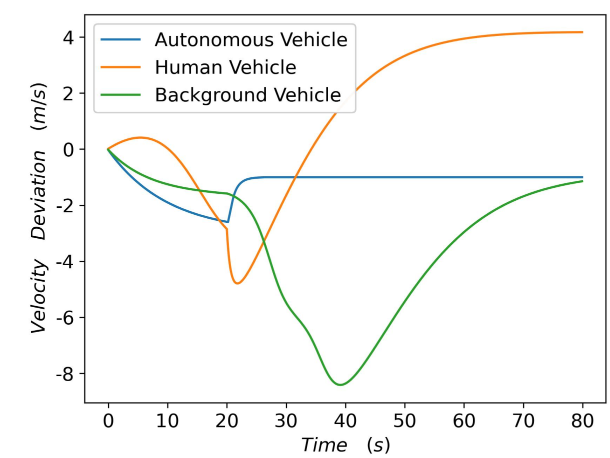


Fig. 4: Velocity Deviation

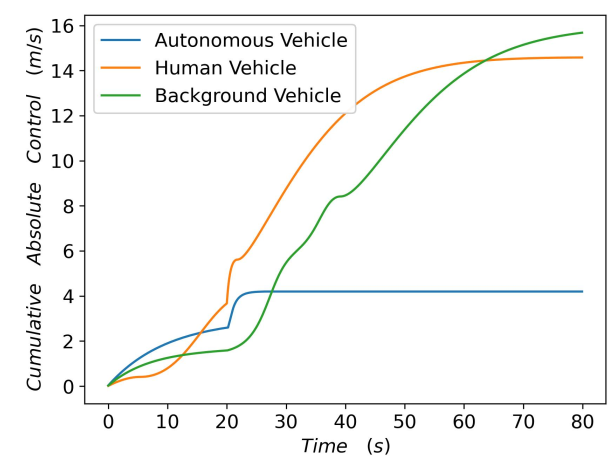
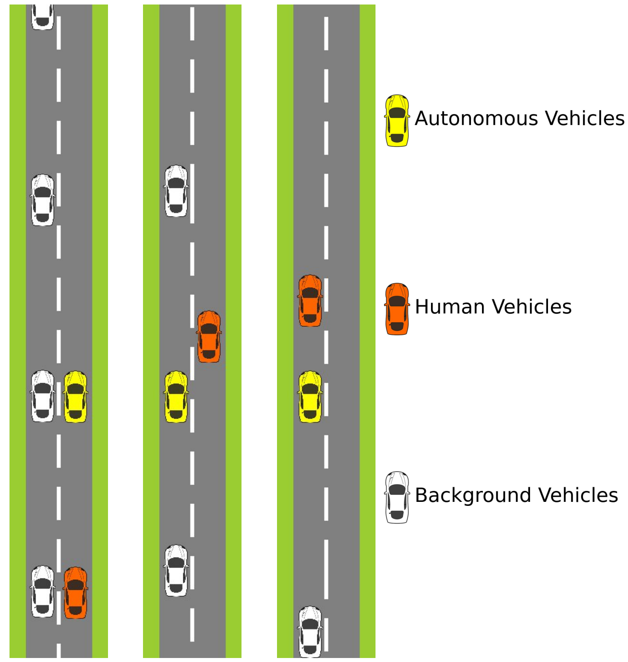


Fig. 5: Cumulative Absolute Control



(a) Phase 1 (b) Phase 2 (c) Phase 3

Fig. 6: Phase 1: Autonomous vehicle merges first. Phase 2: Autonomous vehicle slows down to create gap for human vehicle. Phase 3: Human vehicle merges. All vehicles are traveling upwards.

cle can set a cutoff, $23\,\mathrm{m/s}$ in our simulation for instance, to influence the humans with high desired velocity to drive in the inner lane. According to Fig. 4, the influence brought about 20.04% increase in the human vehicle's velocity, whereas the passive approach wouldn't be able to initiate the influence procedure at all because it does not try to reveal information that the human is not showing, hence the autonomous vehicle becomes more and more wrongly convinced that the human vehicle intends to travel only as fast as $19.86\,\mathrm{m/s}$. According to Fig. 5, the influence introduces a bounded perturbation, about $15.68\,\mathrm{m/s}$ of cumulative absolute control, on average background vehicles, which could be easily attenuated with autonomous vehicles using flow stopper techniques [38], [39].

E. Scenario 2: Helping human to switch lane

Consider a scenario like Fig. 6a, in which the lane the autonomous and human vehicle currently occupy is about to end, either due to traffic, construction, or lane merge. Both vehicles, therefore, have to switch to the left lane, which is occupied by some background vehicles. Assume the headway gaps between the background vehicles are too narrow for humans while traveling at such a high speed. Fortunately, autonomous vehicles are capable of performing the switching. The autonomous vehicle, therefore, helps the human vehicle to switch lanes by first probing the desired headway of the human vehicle around a specific velocity, in this case $20\,\mathrm{m/s}$. The autonomous vehicle will then switch lanes and slow down to create enough gap based on the probed headway (Fig. 6b). Finally, the human vehicle can merge into the lane with ease (Fig. 6c).

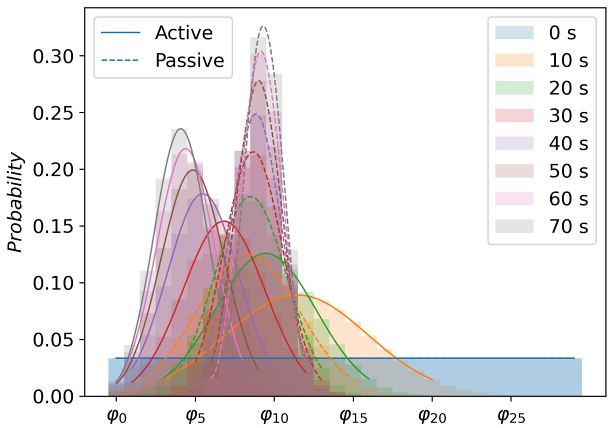


Fig. 7: Belief Snapshot

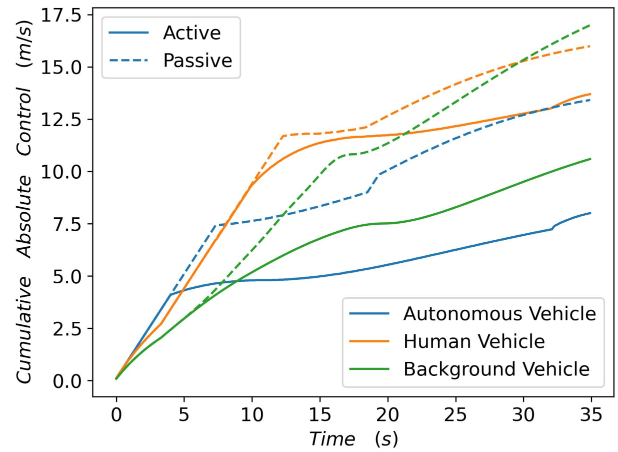


Fig. 9: Cumulative Absolute Control

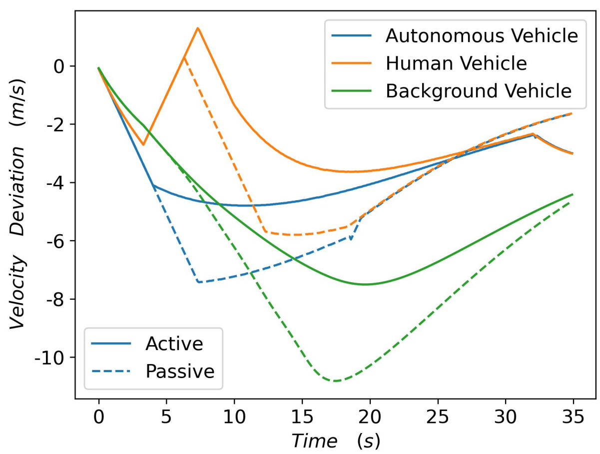


Fig. 8: Velocity Deviation

We choose the IDM parameters as $u_{\max} = 0.73\,\mathrm{m/s^2}$, $b_{\text{pref}} = 1.67\,\mathrm{m/s^2}$, $v_{\text{des}} = 20\,\mathrm{m/s}$, $\tau_{\text{gap}} = 1.5\,\mathrm{s}$, and $d_{\min} = 2\,\mathrm{m}$. Similarly, we initialize the road condition to the same condition as the previous scenario, and we include a passive observing approach to compare as a baseline.

Fig. 7 is a snapshot of the belief from two approaches taken every $10\,\mathrm{s}$. By $70\,\mathrm{s}$, the probability for the active approach peaks at φ_4, which maps to a desirable headway around $48.27\,\mathrm{m}$, whereas that of passive approach peaks at φ_9, which maps to a desirable headway around $108.62\,\mathrm{m}$. For reference, according to data from the Next Generation Simulation for US Highway 101 [40], the average headway for cars traveling around $20\,\mathrm{m/s}$ is about $42.18\,\mathrm{m}$. Although not absolutely precise, the active approach generates a much more accurate profile than the passive approach does.

Based on the probed information, the autonomous vehicle can proceed to create a gap for the human vehicle. For comparison, we simulated a baseline where the autonomous vehicle is passive during the information gathering process, so the autonomous vehicles would have to slow down to create a wider gap, inducing larger perturbations on the background vehicles. According to Fig. 9, the cumulative absolute control for all three types of vehicles in the active approach is significantly lower than that in the passive approach. The reductions in perturbation are respectively 40.36%, 14.33%, and 37.66% for autonomous, human, and background vehicle. According to Fig. 8, the active approach generates less extreme velocity deviation for all three types of vehicles in general, which helps to reduce the intensity and propagation of traffic wave [41].

Moreover, our baseline is under the assumption that the autonomous vehicle would overtake under this scenario. Without active probing, the autonomous vehicle is more likely to behave quite conservatively, so it will most likely wait until all of the background vehicles have passed to switch lanes. This subjects the autonomous and human vehicles to almost a complete stop and a wait time that depends on the number of consecutive closely spaced background vehicles behind, meaning that the deviation will continue to increase if there is no large gap. Our active probing and influencing approach, on the other hand, is agnostic to this condition because the autonomous vehicle creates its own lane-change opportunity.

V. Conclusions

In this work, we present an active probing approach for an autonomous agent to actively interact with a human agent to reveal information about a human's underlying utility and to clarify its belief of the human internal model. Our simulation results in autonomous driving demonstrate how the gathered information can be leveraged to increase driver experience and overall optimality compared to a passive learning baseline method. Future work could include designing optimization that combines probing and generating adversarial trajectories so that no prior learning is needed. It could also be worthwhile to relax the assumption that the human model is static and to empower the autonomous agent to actively learn the human model's adaptation policy.

REFERENCES

[1] H. G. Stassen, G. Johannsen, and N. Moray, "Internal representation, internal model, human performance model and mental workload," *Automatica*, vol. 26, no. 4, pp. 811–820, 1990.

[2] R. J. Jagacinski and R. A. Miller, "Describing the human operator's internal model of a dynamic system," *Human Factors*, vol. 20, no. 4, pp. 425–433, 1978.

[3] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 3931–3936.

[4] M. Liebner, M. Baumann, F. Klanner, and C. Stiller, "Driver intent inference at urban intersections using the intelligent driver model," in *2012 IEEE intelligent vehicles symposium*. IEEE, 2012, pp. 1162–1167.

[5] A. D. Dragan and S. S. Srinivasa, *Formalizing assistive teleoperation*. MIT Press, July, 2012.

[6] M. Awais and D. Henrich, "Human-robot collaboration by intention recognition using probabilistic state machines," in *19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010)*. IEEE, 2010, pp. 75–80.

[7] C. L. Baker, R. Saxe, and J. B. Tenenbaum, "Action understanding as inverse planning," *Cognition*, vol. 113, no. 3, pp. 329–349, 2009.

[8] T.-H. D. Nguyen, D. Hsu, W.-S. Lee, T.-Y. Leong, L. P. Kaelbling, T. Lozano-Perez, and A. H. Grant, "Capir: Collaborative action planning with intention recognition," in *Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2011.

[9] R. D. Luce, *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2012.

[10] P. Abbeel and A. Y. Ng, "Exploration and apprenticeship learning in reinforcement learning," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 1–8.

[11] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," *arXiv preprint arXiv:1206.4617*, 2012.

[12] A. Y. Ng, S. Russell *et al.*, "Algorithms for inverse reinforcement learning." in *Icml*, vol. 1, 2000, p. 2.

[13] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey *et al.*, "Maximum entropy inverse reinforcement learning." in *Aaai*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[14] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning." in *IJCAI*, vol. 7, 2007, pp. 2586–2591.

[15] C.-P. Lam, A. Y. Yang, and S. S. Sastry, "An efficient algorithm for discrete-time hidden mode stochastic hybrid systems," in *2015 European Control Conference (ECC)*. IEEE, 2015, pp. 1212–1218.

[16] D. Kulic and E. A. Croft, "Affective state estimation for human–robot interaction," *IEEE transactions on robotics*, vol. 23, no. 5, pp. 991–1000, 2007.

[17] T. Van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate activity recognition in a home setting," in *Proceedings of the 10th international conference on Ubiquitous computing*, 2008, pp. 1–9.

[18] S. Javdani, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization," *Robotics science and systems: online proceedings*, vol. 2015, 2015.

[19] A. Fern, S. Natarajan, K. Judah, and P. Tadepalli, "A decision-theoretic model of assistance," *Journal of Artificial Intelligence Research*, vol. 50, pp. 71–104, 2014.

[20] T. Bandyopadhyay, K. S. Won, E. Frazzoli, D. Hsu, W. S. Lee, and D. Rus, "Intention-aware motion planning," in *Algorithmic foundations of robotics X*. Springer, 2013, pp. 475–491.

[21] J. F. Fisac, A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, S. Wang, C. J. Tomlin, and A. D. Dragan, "Probabilistically safe robot planning with confidence-based human predictions," *arXiv preprint arXiv:1806.00109*, 2018.

[22] H. Hu, K. Nakamura, and J. F. Fisac, "Sharp: Shielding-aware robust planning for safe and efficient human-robot interaction," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5591–5598, 2022.

[23] R. Tian, L. Sun, A. Bajcsy, M. Tomizuka, and A. D. Dragan, "Safety assurances for human-robot interaction via confidence-aware game-theoretic human models," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 11 229–11 235.

[24] R. P. Bhattacharyya, R. Senanayake, K. Brown, and M. J. Kochenderfer, "Online parameter estimation for human driver behavior prediction," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 301–306.

[25] H. Hu and J. F. Fisac, "Active uncertainty learning for human-robot interaction: An implicit dual control approach," *arXiv preprint arXiv:2202.07720*, 2022.

[26] D. Sadigh, S. S. Sastry, S. A. Seshia, and A. Dragan, "Information gathering actions over human internal state," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 66–73.

[27] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[28] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information theory*, vol. 37, no. 1, pp. 145–151, 1991.

[29] R. Sibson, "Information radius," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 14, no. 2, pp. 149–160, 1969.

[30] S. Kullback, *Information theory and statistics*. Courier Corporation, 1997.

[31] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[32] F. Nielsen, "On a generalization of the jensen–shannon divergence and the jensen–shannon centroid," *Entropy*, vol. 22, no. 2, p. 221, 2020.

[33] R. E. Bellman and S. E. Dreyfus, *Applied dynamic programming*. Princeton university press, 2015, vol. 2050.

[34] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[35] M. Treiber and D. Helbing, "Explanation of observed features of self-organization in traffic flow," *arXiv preprint cond-mat/9901239*, 1999.

[36] M. Treiber and A. Kesting, "Traffic flow dynamics," *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*, pp. 983–1000, 2013.

[37] ——, "The intelligent driver model with stochasticity-new insights into traffic flow oscillations," *Transportation research procedia*, vol. 23, pp. 174–187, 2017.

[38] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S.-i. Tadaki, and S. Yukawa, "Traffic jams without bottlenecks—experimental evidence for the physical mechanism of the formation of a jam," *New journal of physics*, vol. 10, no. 3, p. 033001, 2008.

[39] C. Wu, A. M. Bayen, and A. Mehta, "Stabilizing traffic with autonomous vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6012–6018.

[40] J. Colyar and J. Halkias, "Us highway 101 dataset," *Federal Highway Administration (FHWA), Tech. Rep. FHWA-HRT-07-030*, pp. 27–69, 2007.

[41] A. R. Kreidieh, C. Wu, and A. M. Bayen, "Dissipating stop-and-go waves in closed and open networks via deep reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1475–1480.

Toward Multi-Robot Perception Planning for Filming and Reconstructing Groups of Moving Actors

Skyler Hughes[1] Micah Corah[2] and Sebastian Scherer[2]

***Abstract*— Multi-robot scene reconstruction with multiple moving targets remains a challenging problem in the field of aerial multi-robot planning. Particularly difficult is the task of reconstructing groups of people splitting and merging. State of the art methods often rely on pre-set formations, limiting adaptability in cluttered environments, as well as the ability for the system to explore a wider space of solutions. In addition, state of the art methods typically do not include reconstruction, view quality, and inter-robot collision avoidance into account when planning. This work seeks to formulate the problem of multi-view reconstruction of groups of people as a perception planning problem. We solve this—and achieve suboptimality guarantees—using a combination of value iteration to optimize views for individual robots and sequential submodular maximization methods to coordinate the team. We present a novel submodular objective for cinematography/reconstruction. We demonstrate single robot planning in a simplified setting and provide a planning framework for future use.**

***Index Terms*— Multi-Robot Systems, Submodular maximization, UAV, greedy planning, perception planning**

I. INTRODUCTION

The use of unmanned aerial vehicles (UAVs) has grown continually in recent years, and they are widely useful as mobile sensing and camera platforms. UAVs provide the ability to position a camera anywhere in 3D space, opening up the door to many possibilities across cinematography, inspection, and search and rescue. In many of these applications, it is desirable to acquire multiple view points of the same object at the same time through the use of multiple UAVs. Across these applications the need for teams of robots that are capable of both moving target tracking as well as achieving diversity in view points for cinematic or reconstruction purposes is growing.

Use of UAV's in applications such as filming sports, unscripted performances, and study of animal groups remain challenging for current techniques. Group cinematography for unscripted scenes with groups of moving targets proves difficult due to the challenges in multi-robot coordination, defining appropriate objectives, and system constraints. Modern systems are capable of tracking a single target with a single robot, as well as multiple robots tracking a single target [1], but lack multi-target capability. This has produced systems capable of coordinating multiple robots in cluttered environments that are able to track single targets with diversity in viewpoint for pose reconstruction [1]. Methods like these often rely on formations that simplify the planning process and ensure good performance in canonical situations such as [1], [2]. Attempts to break from these limitations have come in the form of learning based methods [3], which produce focus on keeping a target in view, without a view quality metric. Multi-drone tracking of single actors in highly cluttered environments has been demonstrated [2], but focuses on single target tracking and collision avoidance, not view point diversity or coverage. These systems generally offer limited flexibility in formations and do not yet consider how to optimize camera views for the purpose of high-fidelity reconstruction of moving targets.

In sensor planning schemes submodular maximization has proven to be a powerful technique for solving informative planning problems such reconstruction, coverage, and tracking [4–6], but few works seek to provide realistic sensor models or objectives for the purpose of filming or reconstructing moving targets. Roberts et al. [6] utilize objective submodularity and is capable of global multi-view coverage optimization for static building reconstruction. Corah [4] utilize submodularity and related mathematical properties to derive a number of efficient multi-robot coordination schemes built on top of submodularity [4], but focuses on settings other than reconstruction.

In this work we introduce a method that pairs perception planning methods with cinematic objectives to coordinate multiple UAVs and provide view-diverse tracking objectives for multiple moving targets. We leverage submodularity in our objective formulation along with sequential greedy planning to achieve bounded sub-optimality guarantees on performance. Our framework produces planning solutions that maximize an intuitive *view quality metric* that captures view-diversity, target coverage, and target size reconstruction objectives. Sequential planning and a carefully constructed objective function allow us to avoid exponential complexity in planning common in multi-agent problems.

II. PROBLEM FORMULATION

Consider a team of N robots with states $x_{r,t} \in X_r$ where X_r is a subset of the special euclidean group $SE(2)$ at time t. With a set of actions $u_{r,t} \in U_{r,t}$ where $U_{r,t}$ is the *finite* space of actions available to robot r at time t.

Given a prior distribution of targets (which may be time varying) Y_t, we seek to maximize the *submodular* quality metric ξ Eq. 2, for a sequence of actions over the fixed planning horizon of a L-step look ahead for multiple coordinated

[2] Skyler Hughes is with the Department of Physics and Department of Computer Science at New Mexico Institute of Mining and Technology, NM, USA

[2] Micah Corah and Sebastian Scherer are with the AirLab, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA

This work is supported by the National Science Foundation under grant no. 2024173.

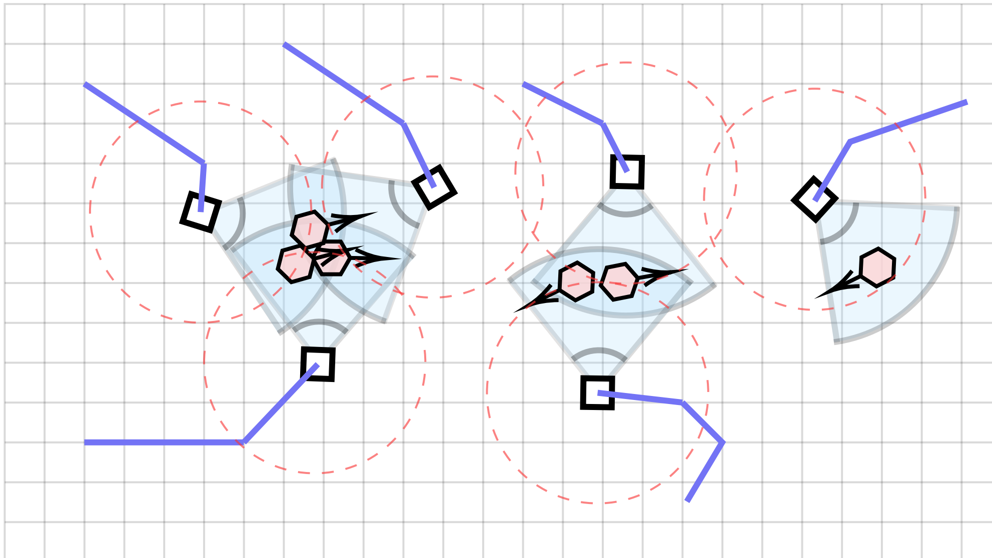


Fig. 1. System concept showing multiple UAVs covering a number of targets from multiple view angles. UAVs are represented with squares, and blue semicircles representing the coverage zones of the sensor model. Blue lines indicate paths through the grid, and red dashed circles indicate the movement range D.

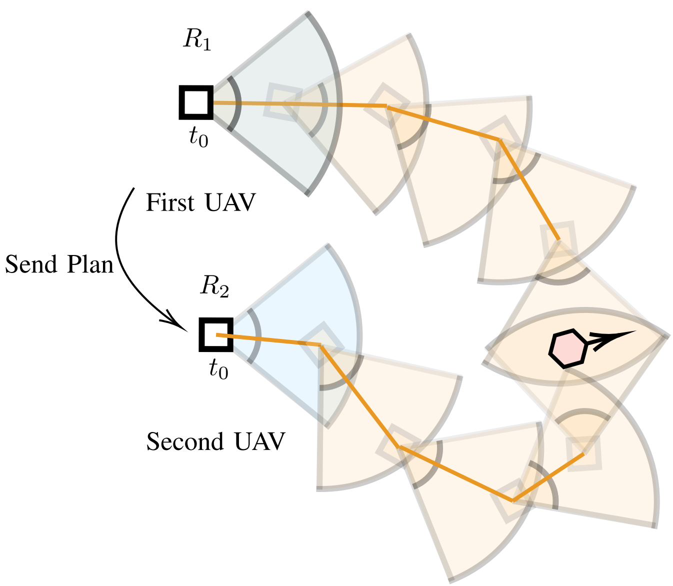


Fig. 2. Sequential planning with submodular cinematic objective. UAV's coordinate to achieve multiple views of the target.

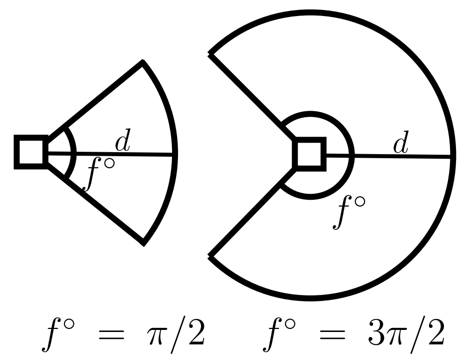


Fig. 3. UAV sensor model, with two different values of f° both facing **east**

robots.

The state space is discretized into a 2D grid and 8 sensor facing angles. We refer to the sensor direction has the *heading* of an agent.

A. Motion Model

State transitions are governed by the following *motion model*.

$$x_{r,t+1} = f_r(x_{r,t}, u_{r,t}) \tag{1}$$

where f_r permits state transitions within euclidean distance D. The transition model also updates the current set of available actions $U_{r,t}$, in order to preserve the distance constraints and boundary conditions. The set of available actions may grow or shrink depending on the location of the state of interest, especially near grid boundaries.

B. Sensor Model

Agents receive observations capable of sensing targets in its environment, and receiving observations based on its current state and the state of the environment. Agents are able to sense targets within a distance d and within angle $f^\circ/2$ of the agent's heading where f° is the field of view of the view cone sensor. Heading is discretized into 8 cardinal directions. Rotation clock-wise(CW) or counter-clock-wise (CCW) constitute the actions capable of changing heading.

An agent with $d = \infty$ and $f^\circ = 360^\circ$ would be able to observe the entire environment at all times. Sensible values of d and f° are chosen based on typical capabilities of real world systems. Fig. 3 showcases the sensor model with two values of f°.

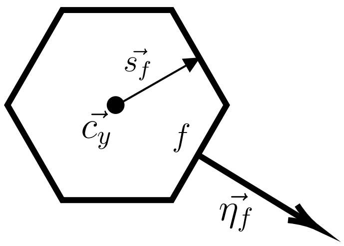


Fig. 4. Target Representation

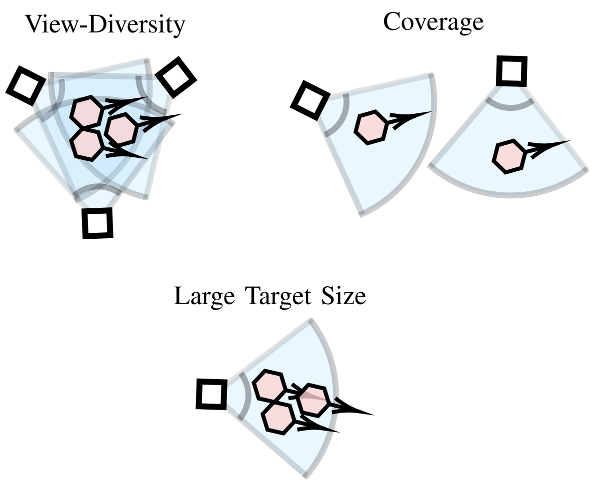


Fig. 5. Summary of desired qualities we capture with our objective.

C. Target Representation

Targets are represented by polygons in the 2D plane. Each target y is parameterized by the tuple $(\vec{c_y}, F_y, s_y)$ where $\vec{c_y}$ is the 2D position of the target, F_y is the set of faces that compose the target, and s_y is the *apothem* of the target. Each face $f \in F_y$ also has associated with it a normal vector $\vec{\eta_f}$ that will be utilized in the objective definition 6. *See Fig.* 4.

D. Objective Functions

We seek to express the following behaviors with our objective formulation.

- View Diversity - UAVs should arrange themselves such that multiple views of the target(s) are visible.
- Maximum target size - UAVs should position themselves such that targets have a large size on the image sensor.
- Maximum target coverage - UAVs should attempt to have all targets in view at all times.

These behaviors are summarized in *Fig.* 5.

We capture these qualities by maximizing the objective Q over a sequence of actions in the finite time horizon $t \in [0...T]$ for all robots, given a prior target distribution (which may be time varying) Y_t. This yields the overall quality metric ξ.

$$\xi = \sum_{t=1}^{T} \max_{u_{r,t} \in U_r,} Q(u_{r,t}) \tag{2}$$

We define objective Q as:

$$Q(u_{r,t}) = \sum_{y_i \in Y_t} \phi_y(u_{r,t}). \tag{3}$$

Where ϕ_y represent the *total view quality* over the set of targets. In order to achieve bounded sub-optimality guarantees afforded by using submodular maximization [5], we seek a *submodular* and *monotonic* objective function. Such a function ϕ is demonstrated in 4:

$$\phi_y(u_{r,t}) = \sum_{f \in F_y} \sqrt{\gamma_f(u_{r,t})} \tag{4}$$

Where γ accumulates the sensing quality for a particular face f of a target across all robots.

Let the distance between the position of the robot P_r and the location of a particular face f on target y be defined:

$$\vec{d_{r,f}} = (\vec{c_y} + \vec{s_f}) - \vec{P_r} \tag{5}$$

$$\gamma_f(U_{r,t}) = \sum_{r \in U_{r,t}} \alpha \text{INVIEW}_f(u_{r,t}) \frac{d_{r,f} \cdot \vec{\eta_f}}{||d_{r,f}||^3} \tag{6}$$

Where $||\cdot||$ is the 2-norm, INVIEW_f returns one if the face f would be detected by taking the action $u_{r,t}$ at time t by robot r, and zero otherwise, and α is the number of pixels per unit area at one meter.[1]

E. Planning Approach

Singh et al. [7] were the some of the first to develop sequential planning methods for multi-robot sensing problems based on submodular maximization. Submodular maximization is a class of approximation algorithms for solving a variety of monotone and submodular maximization problems that show up frequently in robotics. As presented in [4] it can be applied to a wide range of problems that can be modeled by a suitable partition matroid.

We propose a similar architecture as in [4] utilizing sequential greedy assignment for multi-robot coordination. In these algorithms, each robot plans sequentially using an optimal single robot planner to maximize an objective. Coordination in this manner provides performance guarantees for overall system performance given that the single robot planner is optimal and the objective function is submodular and monotonic [4].

III. METHODS

A. Single Robot Planner

We propose the use of a value iteration algorithm to suffice as the single robot planner as previously used in [8, 9]. To accomplish this we model the problem as a markov decision process with the movement and sensor models presented. For our experiments we utilize a simple coverage objective that provides a fixed reward for each target INVIEW. The solver was implemented in Julia using POMDPS.jl [10].

[1]Referring to (4), α will have no effect on the objective except as a scaling factor.

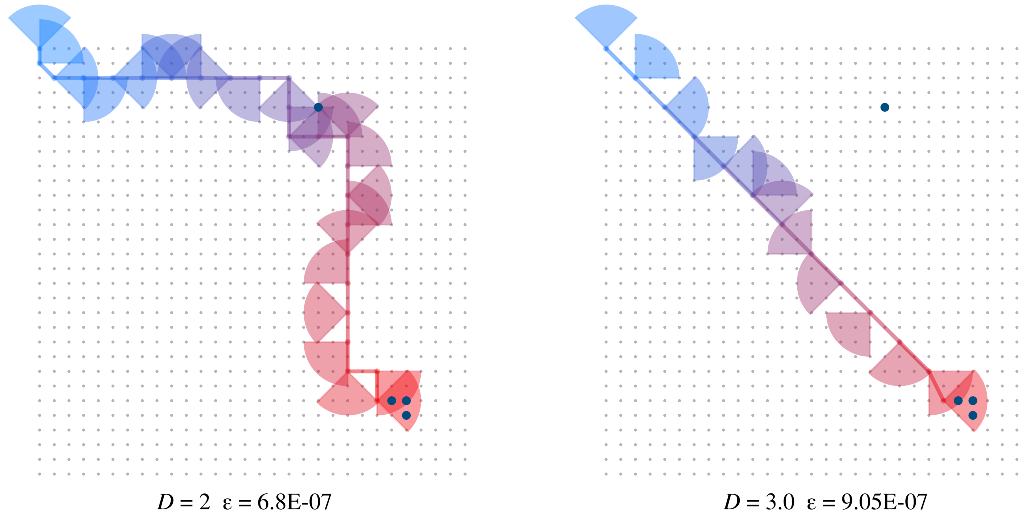


Fig. 6. Move distance comparison. Computed trajectory for a single robot (denoted by view cones). Target locations are represented by large blue circles. We can see that the algorithm achieved a lower Bellman residual in the $D = 2$ case.

Several qualitative experiments were performed to evaluate the trajectories produced by the solver. In all trials the agent starts in the location (1,1) facing north in a 30x30 grid. Static targets are placed in small clusters at various locations throughout the grid. The optimal policy (π^*) is found using the value iteration solver [10], until Bellman residual (ϵ) is less than 1×10^{-6}.

We also compare trajectories for different values of the movement constraint D. *See Fig.* 7.

IV. RESULTS

A. Single Robot Planner

The single robot planner was demonstrated to solve the MDP formulation of the problem and successfully produce valid trajectories as shown in figures 6-7. From this we gather some observations.

- Bellman residuals (ϵ) converged to less than 1×10^{-6} for all experiments.
- Solver converged after 90 iterations on average.
- Computation time increased dramatically with increases in movement distance D.
- Planner correctly prioritizes the largest groups.
- Planner has no provision for total coverage and never leaves highest reward cluster.

V. CONCLUSIONS

We have demonstrated an optimal single robot planner for the motion and sensing models described in a grid setting. The optimality is evidence by the convergence of the value iteration algorithm and the low Bellman residuals. The algorithm demonstrated here satisfies the optimality requirement needed by submodular maximization in order to provide bounded-suboptimality of the entire path [5]. However the algorithm as implemented is computationally intensive, lacks machinery for total target coverage, and does not used the proposed cinematic objective.

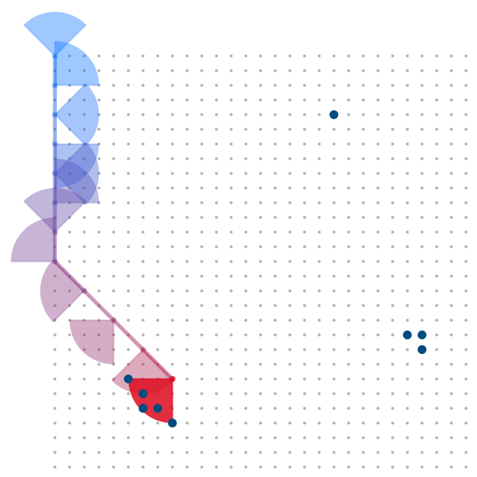


Fig. 7. Computed trajectory for a single robot (denoted by view cones). Target locations are represented by large blue circles. Plan quickly converges to the largest of the available groups, maximizing coverage across the view cone.

In the future, we intend to implement the perception objective described in Sec. II-D as an improvement to the coverage objective demonstrated in the the results. We will also integrate that single-robot planner with greedy multi-robot coordination as proposed. We plan to compare the performance of our method against representative formation planners such as in [1] using reconstruction metrics such as chamfer distance. To facilitate this we will utilize AirSim as a rendering component of the planning pipeline.

References

[1] C. Ho, A. Jong, H. Freeman, R. Rao, R. Bonatti, and S. Scherer, "3d human reconstruction in the wild with collaborative aerial cameras," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep 2021, p. 5263–5269.

[2] X. Zhou, X. Wen, Z. Wang, Y. Gao, H. Li, Q. Wang, T. Yang, H. Lu, Y. Cao, C. Xu, and F. Gao, "Swarm of micro flying robots in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm5954, May 2022. [Online]. Available: https://www.science.org/doi/10.1126/scirobotics.abm5954

[3] R. Tallamraju, N. Saini, E. Bonetto, M. Pabst, Y. Liu, M. Black, and A. Ahmad, "Aircaprl: Autonomous aerial human motion capture using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, 08 2020.

[4] M. Corah, "Sensor planning for large numbers of robots," Ph.D. dissertation, Carnegie Mellon University, 2020.

[5] M. Corah and N. Michael, "Distributed matroid-constrained submodular maximization for multi-robot exploration: Theory and practice," *Autonomous Robots*, vol. 43, no. 2, p. 485–501, 2019.

[6] M. Roberts, S. Shah, D. Dey, A. Truong, S. Sinha, A. Kapoor, P. Hanrahan, and N. Joshi, "Submodular trajectory optimization for aerial 3d scanning," in *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE, Oct 2017, p. 5334–5343. [Online]. Available: http://ieeexplore.ieee.org/document/8237831/

[7] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research*, vol. 34, p. 707–755, Apr 2009. [Online]. Available: https://jair.org/index.php/jair/article/view/10602

[8] A. Bucker, R. Bonatti, and S. Scherer, "Do you see what i see? coordinating multiple aerial cameras for robot cinematography," no. arXiv:2011.05437, Mar 2021, arXiv:2011.05437 [cs]. [Online]. Available: http://arxiv.org/abs/2011.05437

[9] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Decentralized active information acquisition: Theory and application to multi-robot slam," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. Seattle, WA, USA: IEEE, May 2015, p. 4775–4782. [Online]. Available: http://ieeexplore.ieee.org/document/7139863/

[10] M. Egorov, Z. N. Sunberg, E. Balaban, T. A. Wheeler, J. K. Gupta, and M. J. Kochenderfer, "POMDPs.jl: A framework for sequential decision making under uncertainty," *Journal of Machine Learning Research*, vol. 18, no. 26, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-300.html

[11] M. T. J. Spaan, T. S. Veiga, and P. U. Lima, "Decision-theoretic planning under uncertainty with information rewards for active cooperative perception," *Autonomous Agents and Multi-Agent Systems*, vol. 29, no. 6, pp. 1157–1185, 2015.

[12] R. Bonatti, C. Ho, W. Wang, S. Choudhury, and S. Scherer, "Towards a robust aerial cinematography platform: Localizing and tracking moving targets in unstructured environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019, p. 229–236.

[13] E. Hazan, S. M. Kakade, K. Singh, and A. Van Soest, "Provably efficient maximum entropy exploration," no. arXiv:1812.02690, Jan 2019, arXiv:1812.02690 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1812.02690

[14] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 1–43, 2012.

[15] N. Pan, R. Zhang, T. Yang, C. Xu, and F. Gao, "Fast-tracker 2.0: Improving autonomy of aerial tracking with active vision and human location regression," no. arXiv:2103.06522, Mar 2021, arXiv:2103.06522 [cs]. [Online]. Available: http://arxiv.org/abs/2103.06522

[16] S. Foldes and P. L. Hammer, "Submodularity, supermodularity, and higher-order monotonicities of pseudo-boolean functions," *Mathematics of Operations Research*, vol. 30, no. 2, pp. 453–461, 2005.

[17] F. Bach, "Learning with submodular functions: A convex optimization perspective," *Foundations and Trends in Machine Learning*, vol. 6, no. 2-3, 2013.

[18] A. Gupta and R. Levin, "Fully-dynamic submodular cover with bounded recourse," *arXiv preprint arXiv:2009.00800*, 2020.

[19] M. Corah and N. Michael, "Distributed submodular maximization on partition matroids for planning on large sensor networks," in *Proc. of the IEEE Conf. on Decision and Control*, Miami, FL, Dec. 2018.

Spatio-temporal Motion Planning for Autonomous Vehicles with Trapezoidal Corridors and Bézier Curves

Srujan Deolasee[1], Qin Lin[2], and John M. Dolan[3]

***Abstract*— Motion planning in dynamic environments is one of the key modules in autonomous driving systems. Safety-guaranteed motion planning is critical for self-driving cars to generate collision-free trajectories. A layered motion planning approach with decoupled path and speed planning is widely used for this purpose. This approach is prone to be suboptimal in the presence of dynamic obstacles. Spatial-temporal approaches deal with path planning and speed planning simultaneously; however, the existing methods only support simple-shaped corridors like cuboids, which restrict the search space for optimization in complex scenarios. We propose to use trapezoidal prism-shaped corridors for optimization, which significantly enlarges the solution space compared to the existing cuboidal corridors-based method. Finally, a piecewise Bézier curve optimization is conducted in our proposed corridors. This formulation theoretically guarantees the safety of the continuous-time trajectory. We validate the efficiency and effectiveness of the proposed approach in numerical simulations.**

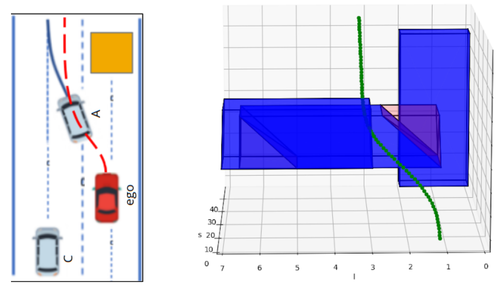
***Index Terms*— Autonomous vehicle navigation, Motion and Path Planning**

I. INTRODUCTION

Autonomous vehicles are promising to revolutionize transportation systems and change how people travel. The vehicle continuously interacts with other agents on the road, like the surrounding cars, pedestrians, ongoing constructions, etc. To interact with these participants, a self-driving car adjusts its path and speed over time based on perception information. This task is modelled using a trajectory optimization problem satisfying safety and dynamic feasibility constraints, while giving importance to comfort. However, solving the original constrained optimization problem is intractable in real-time.

To respond to other road participants and to fulfill the real-time performance requirement, two major trajectory-generation frameworks are proposed in the literature: spatio-temporal planning [1]–[3] and path-speed decoupled planning (also called layered planning) [4]–[6]. These two approaches share the same hierarchical ideas, i.e., finding a heuristic solution as a reference first and optimizing to refine it later.

Spatio-temporal planning considers spatial and temporal maneuvers simultaneously. The spatial parameters are given using two dimensions. A third time dimension is used as



(a) (b)

Fig. 1: Example of yielding scenario and its $S-L-T$ graph

the predicted trajectory is time-profiled and can be generated as a series of spatio-temporal obstacles. Thus, the search and optimization processes are completed in a three-dimensional space. Correspondingly, the layered planning decoupled method decomposes a 3D planning problem into two stages: path planning and speed planning. In the first stage, path planning is executed to generate a path to avoid static, oncoming, and low-speed obstacles. In the second stage, we generate a speed profile along the path to keep a safe distance from dynamic obstacles which block the formed path. It can be argued that the layered planning approach offers more flexibility in both path and speed optimization in the sense that a lot of trajectories are generated for choosing the best option. However, this approach is not optimal with the appearance of dynamic obstacles. Conversely, the direct 3D optimization methods attempt to find the optimal trajectory by theoretically exploring all the convex feasible space. Hence, an optimal trajectory is found even in the presence of dynamic obstacles. The most common approach to speed planning is to use an $S-T$ graph for describing the relationship between station and time. This method does not consider the lateral direction directly, which is why we perform a direct optimization in the 3D $S-L-T$ space.

One of the most important parts of motion planning for autonomous vehicles is ensuring safety in continuous time space, i.e., safety between any two consecutive sampling timesteps must be *guaranteed*. Many existing speed planning methods use discrete time instants to impose safety constraints. However, this technique does not assure safety over the whole planning horizon. While refining the time interval solves this problem, it leads to more decision variables, higher computation costs, and a lack of theoretical guarantee. To address this problem, the spatial corridor (i.e., convex free-space) is widely applied in trajectory generation. We are motivated by these efforts to further extend the spatial

*This work was supported by the National Science Foundation.

[1]Srujan Deolasee is with Department of Computer Science Information Systems,Birla Institute of Technology and Science, Pilani, Rajasthan, India f20191139@pilani.bits-pilani.ac.in

[2] Qin Lin is with the Department of Electrical Engineering and Computer Science, Cleveland State University, Cleveland, OH, the USA q.lin80@csuohio.edu

[3] John M. Dolan is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, the USA jdolan@andrew.cmu.edu

corridor to the spatio-temporal domain to cope better with dynamic obstacles. The convex hull property of Bézier polynomials is leveraged to enforce that the continuous trajectory always falls into a safe region. In addition, such an optimization problem's solution space is enlarged via our proposed trapezoidal-prism shaped corridors.

The main contributions of our work can be briefly described as follows:

1) We propose an efficient convexification algorithm to construct 3D convex-feasible regions consisting of trapezoidal-prism shaped corridors.
2) We provide a sufficient condition on coefficients of the Bézier polynomials to guarantee the trajectory's safety in trapezoidal-prism corridors theoretically. Compared with existing cuboidal corridors [1], the condition is relaxed and the solution space is significantly enlarged, which leads to a higher chance of finding an optimal solution.

The remainder of this paper is structured as follows. We review related works in Sec. II. We introduce necessary notations and background materials on the $S-L-T$ graph for speed planning and Bézier polynomials in Sec. III. The 3D convex safe region construction is presented in Sec. IV. In Sec. V, we present our optimization formulation. The simulation results and analysis can be found in Sec. VI. We make concluding remarks in Sec. VII.

II. Related Works

A. Speed Planning for Autonomous Vehicles

Speed planning techniques can be classified into three categories: 1) search and optimization; 2) sampling lattices and selecting the minimum-cost trajectory; 3) approximated optimization. The first method searches for the best candidate speed profile and optimizes the curve for smoothness. This method is commonly adopted for optimality and efficiency. Xu et al. first presented a method of selecting the best lattice solution and conducting a post-optimization [5]. The Baidu EM motion planner uses dynamic programming for search and piecewise monomial polynomials for optimization [7]. In [8], based on a reference speed profile generated by a heuristic search, the trajectory is optimized using Piecewise-Jerk Speed Optimization with vehicle dynamics as constraints. Yang et al. project the collision-free constraints in the 3D spatio-temporal domain to the 2D space domain [9]. For the second category, different speed lattices are sampled and combined with path lattices. The generated local spatial-temporal trajectories are evaluated and the one with minimum cost is selected. Related works can be found in [4]–[6]. Most works in the first and the second categories conduct search and optimization directly in the $S-T$ graph. For the third category, a vehicle dynamic model is directly considered in a sequential optimization problem. Liu presents a novel slack convex feasible set algorithm [10]. Some control-related optimization methods also fall in this category, such as model predictive control (MPC) [11] and constrained iterative linear quadratic regulator (CiLQR) [12]. The advantage of these approaches is that they mitigate the planning and control inconsistency problem, since the dynamic model has already been considered in the planning layer. However, the disadvantage is the high computation cost. Our proposed planner fulfills the real-time requirement, as discussed further in Sec. VI.

B. Corridor generation for Autonomous Vehicles

The spatial corridor (i.e., convex free-space) is widely applied in trajectory generation. Zhu et al. propose a convex elastic smoothing algorithm which can generate a collision-free "tube" around the initial path [13]. Erlien et al. consider not only spatial information but also vehicle dynamics to construct the convex tube [14]. Both of these works, however, generate the corridor in a static environment and cannot deal with dynamic obstacles. Liu et al. find a convex feasible set around the reference trajectory and leverage the convex feasible set to accelerate the non-convex optimization [15]. However, the computation complexity is still prohibitively high for real-time applications. [16] propose the use of trapezoidal corridors for convexifying 2D space in the $S-T$ graph. Zhang et al. present a general convex spatio-temporal corridors-based approach in [17]. Xu et al. propose using a modified vertical cell decomposition approach for speed planning in [18]. Even though these works consider the presence of dynamic obstacles, they plan the trajectory in a 2D space, which hinders their performance in complex scenarios. Ding et al. use the spatiotemporal semantic corridor (SSC) method to uniformly express obstacles and traffic rules in the 3D $S-L-T$ space [1]. However, restricting the shape of the corridors to simple cuboids drastically limits the search space for optimization in complex scenarios. Our proposed method of extending trapezoid-shaped 2D corridors in $S-T$ to 3D $S-L-T$ space significantly enlarges the solution space for trajectory optimization. This enables us to extend the spatial corridor to the spatio-temporal domain to cope with dynamic obstacles, while meeting the real-time requirement.

C. Bézier Polynomials-Based Planning

In [19], Werling et al. use a quintic monomial polynomial for both the longitudinal and lateral direction based on optimal control theory. However, the quintic monomial polynomial is not suitable for the optimization for the following two reasons: 1) one segment of the polynomial only has limited representation ability and may fail to represent a highly constrained maneuver, and 2) the monomial basis polynomial is not well suited to problems with complex configuration space obstacles and dynamical constraints. In previous works on monomial basis polynomial trajectories [7], [19], the constraints are only enforced/checked on a finite set of sampled points. However, this method may fail to detect collision between the other sampled points, and thus cannot provide any guarantee on safety and feasibility. In [20], a smooth and continuous speed profile is computed by proper curve concatenation without optimization and dynamic obstacles. In the area of unmanned aerial vehicles (UAVs), Bézier polynomials combined with rectangular corridors are widely used [21], [22]. Ding et al. borrow this idea and propose to use similar approaches for motion planning of unmanned ground vehicles (UGVs) [1]. Using the piecewise Bézier curve for two-dimensional trajectory (i.e., the longitudinal direction $s(t)$ and lateral direction $l(t)$) along the reference lane helped mitigate the above-mentioned limitations. The reason for using the piecewise Bézier curve is its convex hull property and hodograph property [21]. However, due to dynamic traffic participants, the safe regions of UGVs are time-varying and different from scenarios considered for UAVs. Accordingly, corridors for Bézier curves are also time-varying, and the common convex hull property does not hold. Therefore, it becomes non-trivial to directly transfer corridor construction methods from UAVs to UGVs. For the S-T graph of UGVs, the boundaries of obstacles are usually straight lines or parabolic curves, since the accelerations of obstacles are usually assumed to be constant over the planning horizon. For UAVs, the obstacles can be represented as circles, rectangles, and polygons, therefore rectangular corridors are easily generated by cube inflation. The challenges lie in generating collision-free convex corridors and enforcing Bézier curves in these time-dependent corridors for safety. In our work, we propose to use time-dependent trapezoidal prism-shaped corridors and give sufficient conditions to enforce Bézier curves in these time-dependent corridors for safety. It is theoretically proved that the trapezoidal prism-shaped corridors can enlarge the solution space for improved optimization.

III. $S-L-T$ GRAPH AND TRAJECTORY REPRESENTATION

In this section, we briefly introduce necessary background materials on the $S-L-T$ graph, Bézier polynomials, and trajectory representations using piecewise Bézier polynomials.

A. $S-L-T$ Graph

Speed planning decides when an autonomous vehicle should reach a point from a planned path. To do so, a speed profile can be generated to map timestamps to space. Werling et al. transformed the problem of structured road trajectory planning from the Cartesian to the Frenet coordinate system [19]. A Frenet frame is a dynamical reference frame constructed from a reference lane along the road. Typically, a route planner is used for generating the reference lane. Since human-like driving behavior can typically be decoupled into lateral movements and longitudinal movements, modeling the free-space in these two directions is a more natural representation than modeling free-space in Cartesian coordinates. This 3D space can be represented with an $S-L-T$ graph which consists of the longitudinal direction S, the lateral direction L, and the time T. The longitudinal and lateral directions are with respect to a Frenet frame. Finding the optimal trajectory in a Frenet frame is essentially a 3D constrained optimization problem.

In the decoupled speed planning approach, $S-T$ and $S-L$ graphs have been used to depict a scenario on the road. The $S-L$ graph describes the longitudinal coordinate against the lateral coordinate, while the $S-T$ graph illustrates the change of longitude with respect to time. The vehicle's trajectory is then determined by combining both the graphs. Similarly, the $S-L-T$ graph represents all traffic participants' positions, including the ego vehicle and the surrounding vehicles, simultaneously in all three dimensions. The predicted stations of dynamic obstacles can also be projected.

B. Representing Dynamic Agents in $S-L-T$ graph

As an example, in Fig. 1a, we take the case of two dynamic obstacles moving with constant speeds for simplicity. The scenario is described as follows: Vehicle A and the ego vehicle are driving in a lane which has a static obstacle (e.g.: a construction site) in its lane after some distance along the road. Thus, both of them need to change their lanes. The car C, however, is in a safe lane and continues driving straight with constant velocity. We assume that car A has constant longitudinal and lateral velocities for simplicity. We observe that the blue volume denoting each obstacle in the $S-L-T$ graph is a parallelepiped having a non-zero slope 1) in four planes if lateral velocity is non-zero (e.g.: car A), and 2) in two planes if lateral velocity is zero (e.g.: car C). The side length of the parallelogram along the S axis is equal to the length of the vehicle plus half the length of the ego vehicle to give a safety region, and that along the L axis is the width of the vehicle plus the safety region. The station profile of the ego vehicle on the $S-L-T$ graph reflects its distances from obstacles with respect to time and its decisions such as yielding, overtaking, following, etc. To ensure safety, the feasible space of the speed curve should not have any overlap with the regions projected by obstacles. The solution space is non-convex in general. Constructing corridors in case 1 is challenging, as four faces have non-zero slope.

In this work, we propose an over-approximation of the obstacles like car A in Fig. 1b, and representing them as perfect parallelepipeds, as in obstacles like car C. To prevent any compromise of safety, the over-approximation adds some volume on any two opposite faces having non-zero slopes so as to make their slopes zero in the newly constructed polyhedron. The faces chosen for the over-approximation step are decided by a simple minimization of the volume of search space compromised in the process. Intuitively, if the lateral velocity is less than the longitudinal velocity of the vehicle, the corresponding faces are chosen for over-approximation, as can be seen in Fig. 2. The over-approximation enables us to leverage 2D corridor construction algorithms and extend them to 3D. This ensures the complexity of the algorithm is low enough for the algorithm to be feasible for real-time planning. More details about this process are covered in Sec. IV.

C. Bézier Polynomials and Properties

A Bézier polynomial is a polynomial function represented by linear combinations of Bernstein bases. The nth-order

Bézier polynomial is written as

$$B(t) = c_0 b_n^0(t) + c_1 b_n^1(t) + \cdots + c_n b_n^n(t) = \sum_{i=0}^{n} c_i b_n^i(t)$$

where the Bernstein bases satisfy $b_n^i(t) = C_i^n \cdot t^i \cdot (1-t)^{n-i}, t \in [0,1]$. The coefficients of the polynomial $c_i (i = 0, 1, \ldots, n)$ are also called control points. Compared to monomial polynomials, Bézier curves have the following properties:

- The time interval is defined on $t \in [0,1]$.
- The Bézier polynomial starts at control point $B(0) = c_0$ and ends at $B(1) = c_n$.
- **Convex hull property:** The Bézier curve $B(t)$ is confined within the convex hull of control points. The convex hull property is suitable for the problem of constraining the curve in a convex free-space. Specifically, the Bézier curve $B(t))$ is guaranteed to be entirely confined in the convex hull supported by the control points $\mathbf{c}$. In other words, by constraining $\mathbf{c}$ inside the convex free-space, the resulting curve is guaranteed to be collision-free.
- **Hodograph property:** The hodograph property facilitates constraining high-order derivatives of the Bézier curve, which is useful for enforcing dynamical constraints. By the hodograph property, the derivative of $B(t), \dot{B}(t)$, can also be written as a Bézier polynomial with control points $c_i^1 = n \cdot (c_{i+1} - c_i), i = 0, 1, \ldots, n-1$. In this way, we are also able to calculate arbitrary derivatives of $B(t)$. Similarly, control points of $\frac{\mathrm{d}^{l+1} B(t)}{\mathrm{d} t^{l+1}}$ and $\frac{\mathrm{d}^{l} B(t)}{\mathrm{d} t^{l}}$ satisfy $c_i^{l+1} = (n-l)\left(c_{i+1}^l - c_i^l\right)$. By applying the convex hull property to the derivative Bézier curve, the entire dynamical profile of the original curve $B(t)$ can be confined within a given dynamical range.

D. Trajectory Representation using Bézier Polynomials

To mitigate the numerical instability issue, piecewise Bézier polynomials with lower orders are used instead of using a high-order Bézier polynomial for the whole planning horizon. Each piece of the trajectory is associated with one trapezoidal-prism corridor. Note that $B(t)$ is defined on a fixed time interval $[0,1]$. For a whole trajectory with $m+1$ pieces, in each piece $[T_k, T_{k+1}]$ $(k = 0, 1, \ldots, m)$, we use a scaling transformation and translation transformation in the time domain to map it into the interval $[0,1]$ [13]. Then, the whole piece-wise trajectory in one dimension $\sigma \in \{s, l\}$ can be represented as

$$f^{\sigma}(t) = \begin{cases} h_0 B_0\left(\frac{t-T_0}{h_0}\right), t \in [0, T_1] \\ h_1 B_1\left(\frac{t-T_1}{h_1}\right), t \in [T_1, T_2] \\ \vdots \\ h_m B_m\left(\frac{t-T_m}{h_m}\right), t \in [T_m, T_{m+1}]. \end{cases}$$

where h_i is the scaling transformation factor and T_i is the translation transformation factor for $i = 0, 1, \ldots, m$ with setting $T_0 = 0$.

IV. CORRIDOR GENERATION

Since the speed optimization problem is non-convex, it is infeasible to solve it directly during online planning. As explained before, a convexification algorithm is needed to construct convex corridors for solving the optimization. In this section, we cover the corridor construction procedure in greater details. A reference trajectory is often used to provide a warm start to the optimization process. Having a feasible reference trajectory is important for the next optimization problem as we penalize the deviation from the references in our cost function. In this work, we use simple piecewise functions for generating valid reference waypoints in the configuration space of the ego vehicle.

A. Piecewise Convex Safe Regions Representations

One of the main challenges of motion planning is that the free space is nonconvex. Suppose the whole safe region is divided into $m+1$ pieces with time intervals $[T_0, T_1], \ldots, [T_m, T]$ and $T = T_{m+1}$, with each interval corresponding to a convex safe region. The details of such a convexification algorithm will be introduced in the next section. The k-th convex safe region in $S-L-T$ space can be represented as

$$\begin{gathered} \mathcal{S}_k = \{\ (t_i, s_i, l_i) \mid \\ \underline{p_0^k} + h_k \underline{p_1^k} \frac{t_i - T_k}{h_k} \leq s_i \leq \overline{p_0^k} + h_k \overline{p_1^k} \frac{t_i - T_k}{h_k}, \\ l_{beg} \leq l_i \leq l_{end}, \\ t_i \in [T_k, T_{k+1}]\ \} \end{gathered}$$

where s_i and l_i are the longitudinal and lateral coordinates of the i^{th} control point respectively, $\underline{p_0^k}, \underline{p_1^k}$ are bias and skew of the lower bound and $\overline{p_0^k}, \overline{p_1^k}$ are those of the upper bound. h_k denotes the length of the k-th time interval and satisfies $h_k = T_{k+1} - T_k, k = 0, 1, \ldots, m$.

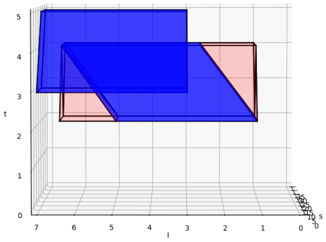
Then, the whole safe region is the union of a set of piecewise-safe sub-regions: $\mathcal{S} = \mathcal{S}_0 \cup \cdots \cup \mathcal{S}_m$. The speed planning is safe if $\forall t_0 \in [0, T], s(t_0) \in \mathcal{S}, l(t_0) \in \mathcal{S}$, which is equivalent to for $t_0 \in [T_k, T_{k+1}], s(t_0) \in \mathcal{S}_k, l(t_0) \in \mathcal{S}_k, k = 0, 1, \ldots, m$, i.e.,

$$\begin{gathered} \underline{p_0^k} + h_k \underline{p_1^k} \frac{t_0 - T_k}{h_k} \leq s(t_0) \leq \overline{p_0^k} + h_k \overline{p_1^k} \frac{t_0 - T_k}{h_k} \\ l_{beg} \leq l(t_0) \leq l_{end} \end{gathered}$$

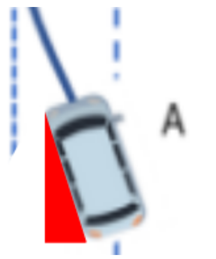
B. Construction of Piecewise-Convex Safe Regions

Algo. 1 outlines the 3D trapezoidal corridor generation process. The original non-convex space with over-approximated dynamic agents is sliced along the L axis at the starting or ending L coordinates of any obstacles in the $S-L-T$ graph. This gives us 3D chunks of the non-convex space which can be projected in a 2D $S-T$ graph *without* the loss of any search space. E.g.: In Fig. 2a, any slices at L coordinates in the range $[1,3)$ will give us the 2D $S-T$ cross section as seen in Fig. 2c. Similarly, any slice at an L coordinate between $[3,6.7)$ will give us the $S-T$ graph as

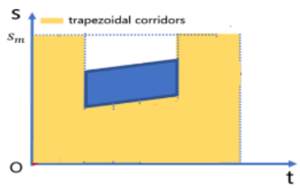
< TABLE OF CONTENTS



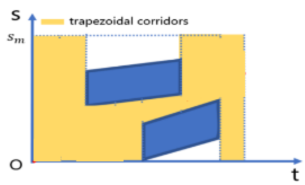
(a) $S-L-T$ graph



(b) bird's eye view with over-approximation (red)



(c) $S-T$ graph for $l \in [1, 3)$



(d) $S-T$ graph for $l \in [3, 6.7)$

Fig. 2: Scenario in Fig. 1b after over-approximating car A

seen in Fig. 2d. The inputs to Algo. 1 are an array of upper and lower bounds in the S and L direction with respect to the ego vehicle and the road. The size of these arrays (i.e., the number of slices) depends on the number of obstacles in the environment and their representation in the $S-L-T$ graph. For each slice, we have lower and upper bounds on the S axis with respect to the ego vehicle. They are measured over a time horizon with discrete time interval Δ. For each slice, we construct 2D convex corridors in the corresponding $S-T$ graph. In this work, we adopt a trapezoidal corridor generation algorithm [16] for constructing the 2D convex corridors. The modified algorithm is presented as Algo. 2.

Algo. 2 outlines the construction of 2D piecewise-convex safe regions in any given $S-T$ cross-section along the L axis. The lower and upper bounds in the S direction serve as inputs to this algorithm. We refer readers to [16] for further details about the working of Algo. 2. As a subroutine in Algo. 3, $SingleRegionCalculate()$ computes the region's bias and skews of the lower bound and the upper bounds. In the loop of Algo. 2 (Lines 4-15), two consecutive meta-pieces are evaluated iteratively. If they can form a new skew (of upper bound or lower bound) that differs significantly from the previous region (see the condition in Line 7, Algo. 2), a new single region will be established. A key modification in our work is that in the subroutine $SingleRegionCaculate()$, we also initialize the upper and lower boundaries of the regions in the L direction (Algo. 3. Lines 6,7). Our over-approximation step and the design of Algorithm 1 guarantee that these boundaries are the same for all 2D convex regions generated by algorithm 2. Thus, we essentially get 2D trapezoidal-shaped corridors *dragged* along the L axis to form 3D trapezoidal prism-shaped convex corridors.

Finally, $RegionSplit()$ is used to check the length of each 2D convex region. If it is above a user-defined threshold (e.g., 1 s in our experimental setting), it will be split into multiple sub-regions, for which the time intervals are all below the threshold. This refinement operation aims to avoid underfitting. Conversely, the abstraction operation $RegionMerge()$ aims to merge the small regions into a larger one, which aims to avoid overfitting and speed up the process.

This step of 2D corridor generation is repeated for all distinct boundaries in our $S-L-T$ graph (Line 2, Algo. 1). The initialization of bounds along the L axis ensures that we get 3D trapezoidal-shaped convex corridors. Since the length of the corridors in the L direction is given by the starting or ending of the obstacles in the $S-L-T$ space, we can guarantee the safety of all the corridors generated using Algo. 1. Note that for the space divided by obstacles, we select the unique space enclosing the comfort-optimal reference trajectory. This step is performed by the $SelectCorridors()$ method. This completes our 3D corridor generation process. Constructing trapezoidal corridors in the 2D cross-sections of our $S-L-T$ graph ensures that the 3D corridors have enlarged solution space as compared to the cuboidal corridor generation presented in [1].

Algorithm 1: Piecewise 3D Convex Regions Generation

Input:
$obs, lb_s[obs], up_s[obs], lb_l[obs], up_l[obs], nums, \Delta$
Output: 3D_Region
Initialize: for $i \leftarrow 0$ ***to*** obs **do**
  $corridor =$
  $Convexify2D(lb_s[i], ub_s[i], lb_l[i], ub_l[i], nums, \Delta)$
  $new\_corridor.append(corridor)$
end
$final\_corridor = SelectCorridors(new\_corridor)$
Return $final\_corridor$

V. PIECEWISE BÉZIER POLYNOMIAL OPTIMIZATION

In this section, we discuss more about the limitations of using the cuboidal corridors. We then discuss the safety enforcement in our trapezoidal prism-shaped corridors. The formulation of quadratic optimization using the newly designed convex solution space is introduced thereafter.

A. Limitations of Safety Enforcement in Cuboidal Corridors

As discussed previously in Sec. III C, the convex hull property of the Bézier curves is used to enforce that the trajectory in the $S-L-T$ graph stays in the safe region $\mathcal{S}$. We first formally define a corridor for our trajectory generation:

Definition V.1. *Let the coefficients of the Bézier Polynomial be* $c_i \in \Omega, i = 0, 1, \ldots, n$. *Here, each control point has two dimensions -* $\{S, L\}$. *These control points lying in the safe region* $\mathcal{S}$ *form a subset* $\mathcal{S}^{cor} \subseteq \mathcal{S}$. *Then, the subset is called a corridor.*

Algorithm 2: Convexify2D

Input: $lb_s, ub_s, lb_l, ub_l, nums, \Delta$
Output: $regions$
Initialize: $regions[0], i = 0, j = 1$ `/* i and j are counters for meta-pieces and resulting convex regions, respectively */`
$SingleRegionCalculate(region, 0, lb_s[0], ub_s[0], lb_s[1], ub_s[1], lb_l, ub_l)$
$regions.append(region)$
for $i \leftarrow 2$ ***to*** $nums - 1$ **do**
  $lskew = (lb_s[i] - lb_s[i-1])/\Delta$ `/* lower bound's skew for two consecutive meta-pieces */`
  $uskew = (ub_s[i] - ub_s[i-1])/\Delta$ `/* upper bound's skew */`
  if $||lskew - regions[j-1].lskew|| > \epsilon$ ***or*** $||uskew - regions[j-1].uskew|| > \epsilon$ **then**
    $regions[j-1].t_{end} = i$
    $regions[j-1].t = (regions[j-1].t_{end} - regions[j-1].t_{beg}) * \Delta$
    $SingleRegionCalculate(region, j, lb_s[i], ub_s[i], lb_s[i+1], ub_s[i+1], lb_l, ub_l)$
    $regions.append(region)$
    $j \leftarrow j + 1$
  end
end
$regions[j-1].t_{end} = nums - 1$
$regions[j-1].t = (regions[j-1].t_{end} - regions[j-1].t_{beg}) * \Delta$
$RegionSplit(regions)$
$RegionMerge(regions)$
Return $regions$

Algorithm 3: Single Region Caculate

Input:
$region, j, lb_s[i], ub_s[i], lb_s[i+1], ub_s[i+1], lb_l, ub_l$
Output: $region$ `/* updated region */`
$region.t_{beg} = j$
$region.lskew = (lb_s[i+1] - lb_s[i])/\Delta$
$region.lbias = lb_s[i]$
$region.uskew = (ub_s[i+1] - ub_s[i])/\Delta$
$region.ubias = ub_s[i]$
$region.l_{beg} = lb_l$
$region.l_{end} = ub_l$

Ding et al. presented the construction of cuboidal corridors in the 3D $S-L-T$ graph in [1]. Constraints of the control points of cuboidal corridors are given by the following proposition:

Proposition V.1. *If a trajectory has control points in each time interval satisfying* $c_i^k \in \Omega^k_{cub} = \{c^k | \underline{p_0^k} + h_k \underline{p_1^k} \le c^{k,s} \le \overline{p_0^k}, l^k_{beg} \le c^{k,l} \le l^k_{end}, i = 0, 1, \ldots, n, k = 0, 1, \ldots, m\}$, $f(t)$ *is guaranteed to be safe, and the upper bounds and lower bounds form cuboidal corridors* $\mathcal{S}^{cub}$.

The proof of safety enforcement in rectangular corridors can be found in [16], and can be easily extended to another dimension L.

The problem of using cuboidal corridors in the $S-L-T$ graph is that when the bounds of corridors in the S direction meet $\underline{p_0^k} + h_k \underline{p_1^k} > \overline{p_0^k}$, there is no feasible solution of the optimization problem and the planner will fail. In order to avoid this case, the time intervals of the k-th corridors should satisfy $h_k \le \frac{\overline{p_0^k} - \underline{p_0^k}}{\underline{p_1^k}}$.

In [1], Ding et al. proposes a seed generation and cube inflation method to adjust time intervals. However, this method will generate a significant number of corridors and optimized variables, which leads to a high computation cost. In addition, if $\exists\ \underline{p_1^k} > 0$ or $\overline{p_1^k} > 0$ the safe regions are not fully covered by cuboidal corridors. As a result, constraints on control points to enforce the station curve in cuboidal corridors are overtightened with reduced solution space (see the illustrations in Fig. 3(a) and 3(c) for examples of this in 2D). In the next subsection, we will introduce the safety enforcement for our proposed trapezoidal-prism corridors with enlarged solution space.

B. Safety Enforcement in Trapezoidal-Prism Corridors

The sufficient conditions of control points c_i to keep the longitudinal and the lateral trajectory safe and in our proposed trapezoidal-prism corridors are built upon the following lemma.

Lemma V.1. *Let* $M \in \mathbb{R}^{(n+1)\times(n+1)}$ *denote the transition matrix from the Bernstein basis* $\{b_n^0(t), b_n^1(t), \ldots, b_n^n(t)\}$ *to the monomial basis* $\{1, t, t^2, \ldots, t^n\}$. *We have* $M_{i,0} = 1, 0 \le M_{i,j} \le 1, i = 0, 1, \ldots, n, j = 0, 1, \ldots, n$.

The proof can be found in [16].

We leverage the following theorem meant for 2D trapezoidal corridors to construct 3D trapezoidal prism-shaped corridors.

Theorem V.1. *For a trajectory, if it has control points in each time interval satisfying* $c_i^k \in \Omega^k$, *where* $\Omega^k = \{c^k | \underline{p_0^k} + h_k \underline{p_1^k} M_{i,1} \underline{p_1^k} \le c_i^{k,s} \le \overline{p_0^k} + h_k \overline{p_1^k} M_{i,1}, l^k_{beg} \le c_i^{k,l} \le l^k_{end}, i = 0, 1, \ldots, n, k = 0, 1, \ldots, m\}$, $f(t)$ *is guaranteed to be safe. The upper and lower bounds in the* S *and* L *directions help form a trapezoidal prism-shaped corridor* $\mathcal{S}^{trp}$.

The proof for the 2D case of the above theorem can be found in [16], and can be easily extended to another dimension L.

In Theorem 1, conditions on c_i^s are $\underline{p_0^k} + h_k \underline{p_1^k} M_{i,1} \underline{p_1^k} \le c_i^{k,s} \le \overline{p_0^k} + h_k \overline{p_1^k} M_{i,1}$. Compared to the safety enforcement in cuboidal corridors in Proposition 1, we have $\underline{p_0^k} + h_k \underline{p_1^k} M_{i,1} \le \underline{p_0^k} + h_k \underline{p_1^k}$ and $\overline{p_0^k} + h_k \overline{p_1^k} M_{i,1} \ge \overline{p_0^k}$. The advantage of having trapezoidal corridors is twofold: i) By the proof of $\underline{p_0^k} + h_k \underline{p_1^k} M_{i,1} \underline{p_1^k} \le c_i^{k,s} \le \overline{p_0^k} + h_k \overline{p_1^k} M_{i,1}$, the lower boundaries are guaranteed to be smaller than the upper boundaries all the time. Recall that for the rectangular corridors, we need to always check $h_k \leq \frac{\overline{p_0^k}-p_0^k}{p_1^k}$; ii) The constraints are relaxed, therefore the solution space is enlarged compared with the rectangular corridors (see the illustration for the comparison in Fig. 3).

C. Trajectory Optimization Formulation

The objective function is established as

$$
\begin{aligned}
J &= J_s + J_l \\
J_s &= w_1 \int_0^T \left(s(t) - s^r(t)\right)^2 \, \mathrm{d}t + w_2 \int_0^T \left(\dot{s}(t) - v_s^r\right)^2 \, \mathrm{d}t \\
&+ w_3 \int_0^T \ddot{s}(t)^2 \, \mathrm{d}t + w_4 \int_0^T \dddot{s}(t)^2 \, \mathrm{d}t + w_5 \left(s(T) - s^r(T)\right)^2 \\
J_l &= w_6 \int_0^T \left(l(t) - l^r(t)\right)^2 \, \mathrm{d}t + w_7 \int_0^T \left(\dot{l}(t) - v_l^r\right)^2 \, \mathrm{d}t \\
&+ w_8 \int_0^T \ddot{l}(t)^2 \, \mathrm{d}t + w_9 \int_0^T \dddot{l}(t)^2 \, \mathrm{d}t + w_{10} \left(l(T) - l^r(T)\right)^2
\end{aligned} \tag{1}
$$

where $s^r(t)$ and $l^r(t)$ are the reference longitudinal and lateral trajectories, and v_s^r and v_l^r are the reference velocities in the two directions. For J_s and J_l: The first term penalizes the deviation from the reference. The second one penalizes the deviation between the actual and reference speed. The third and fourth terms penalize acceleration and jerk, respectively. The last term penalizes the deviation of the ending station from the reference. We used Optuna [23] for tuning all the 10 parameters for one scenario. These weights were further adjusted manually to solve the optimization of all the various scenarios tested.

The optimization considers the following constraints:

- Boundary Constraints: The piecewise curve starts from fixed position, speed, and acceleration, i.e.,

$$c_i^{0,l} h_k^{(1-l)} = \left.\frac{\mathrm{d}^l f(t)}{\mathrm{d}t^l}\right|_{t=0}, \quad l = 0, 1, 2,$$

  where $c_i^{k,l}$ is the control point for the lth-order derivative of the k-th Bézier curve. Note that $c_i^{k,l}$ has two dimensions: $\{S, L\}$.
- Continuity Constraints: The piecewise curve must be continuous at the connected time points for position, speed, and acceleration.

$$c_n^{k,l} h_k^{(1-l)} = c_0^{k+1,l} h_{k+1}^{(1-l)}, l = 0, 1, 2, k = 0, 1, \ldots, m-1$$

- Safety Constraints: With our proposed trapezoidal-prism corridors, safety constraints for the longitudinal dimension of the control point can be given as

$$\underline{p_0^k} + h_k \underline{p_1^k} M_{i,1} \leq c_i^{k,0} \leq \overline{p_0^k} + h_k \overline{p_1^k} M_{i,1}, k = 0, 1, \ldots, m$$

  and those for the lateral dimension of the control point can be given as

$$l_{beg} \leq c_i^{k,0} \leq l_{end}$$

- Physical Constraints: The physical constraints under consideration include the limit of a vehicle's velocity, acceleration, and jerk. We can use the hodograph property of a Bézier curve to calculate velocity, acceleration, and jerk. The constraints are given by

$$
\begin{aligned}
\underline{\beta^{k,1}} &\leq c_i^{k,l} \leq \overline{\beta^{k,1}} \\
\underline{\beta^{l}} &\leq c_i^{k,l} \leq \overline{\beta^{l}}, l = 2, 3
\end{aligned}
$$

  where $k = 0, 1, \ldots, m$ and it follows that $c_i^{k,l+1} = (n-l)\left(c_{i+1}^{k,l} - c_i^{k,l}\right)$. The upper bounds $\overline{\beta^{k,1}}$ are determined by speed limits on road and centripetal acceleration constraints. Let a_{cm} be the maximum acceleration permitted and κ_k the maximum curvature of the path for $t \in [T_k, T_{k+1}]$ (see [24] for details). The lateral acceleration constraints are given by

$$c_i^{k,l} \leq \overline{\beta^{k,1}} = \sqrt{\frac{a_{cm}}{\kappa_k}}.$$

  The bounds on longitudinal and lateral accelerations and jerks are constant for different pieces of speed profiles.

Then, the trajectory optimization process can be formulated as a quadratic programming (QP) problem as

$$
\begin{aligned}
\mathbf{P}: \quad & \min_{\mathbf{c}} \frac{1}{2} \mathbf{c}^T \mathbf{Q_c} \mathbf{c} + \mathbf{q_c}^T \mathbf{c} + \text{const} \\
& \text{s.t. } \mathbf{A}_{eq} \mathbf{c} = \mathbf{b}_{eq} \\
& \quad\quad \mathbf{A}_{ie} \mathbf{c} \leq \mathbf{b}_{ie}.
\end{aligned}
$$

We refer readers to the appendix for the detailed formulation process. This problem can be solved in real-time by a modern solver such as OSQP [25].

VI. SIMULATIONS AND RESULTS ANALYSIS

The experiments done to validate our approach assume that the poses of the surrounding vehicles can be predicted. Our framework has been implemented using C++11. All simulations are carried out on a personal computer with a quad-core 2.60GHz Intel i10-10750H processor.

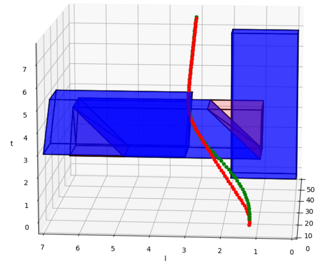
A. Numerical Simulations

We conduct numerical simulations to validate the optimality and the low failure rate of the proposed approach compared to Bézier polynomials with cuboidal corridors. The planning horizon is 7 s. Different road scenarios are considered as follows:

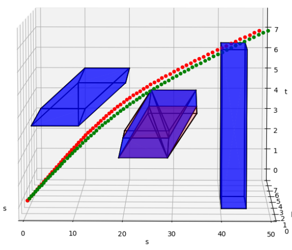
1) Merging into another lane due to road construction: Consider the scenario in Fig. 1. We project different stations of the vehicles onto the $S-L-T$ graph. The initial velocity and the acceleration of the ego vehicle are $v_s(0) = 7.0\ m/s, v_l(0) = 0.0\ m/s$ and $a_s(0) = 0\ m/s^2, a_l(0) = 0\ m/s^2$, respectively.

Fig. 3a and Fig. 3b show Bézier curves generated by using cuboidal (red) and trapezoidal-prism (green) corridors for the scenario presented in Fig. 1. Although both the trajectories look similar, acceleration plots in longitudinal and lateral direction can be used to compare the two solutions. From Fig. 3e, we can conclude that the maximum acceleration required for our method is less than that needed by the cuboidal corridors approach. The superiority of using

< TABLE OF CONTENTS

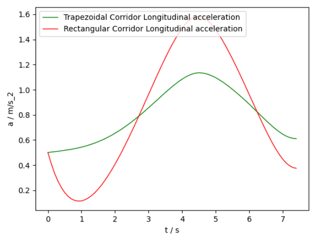


(a)

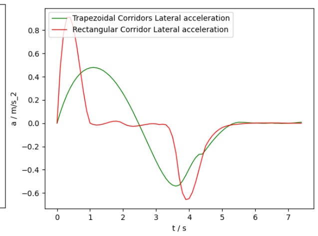


(b)

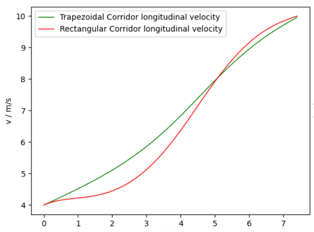
Generated trajectory using both methods (green: trapezoidal, red: cuboidal)



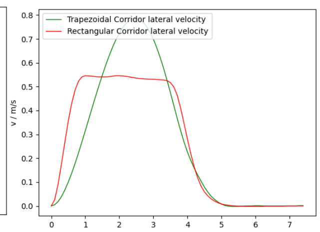
(c) Longitudinal Acceleration profiles



(d) Lateral Acceleration profiles



(e) Longitudinal Velocity profiles

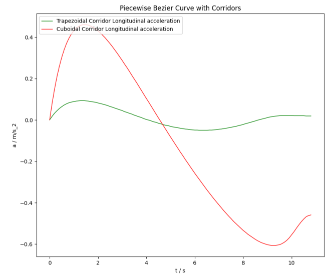


(f) Lateral Velocity profiles

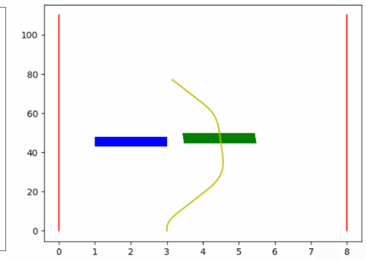
Fig. 3: Piecewise Bézier polynomial and its dynamic profile

trapezoidal corridors is more clear from Fig. 3c, which records the lateral acceleration of both the methods. We observe that our approach yields a smoother acceleration plot with minimal jerk and the lower maximum acceleration. We also test the maximum initial conditions of both the methods for the same scenario to show the effect of the enlarged search space. While using trapezoidal corridors, we could generate a trajectory for $a_s = 2\ m/s^2, v_s = 10.5\ m/s, a_l = 1.2\ m/s^2, v_l = 2\ m/s$ where the bounds on longitudinal acceleration were $[-3.0, 2.0] m/s^2$ and those on lateral acceleration were $[-2.0, 2.0]\ m/s^2$. Using the cuboidal corridors method failed to generate a trajectory for these initial conditions and was only successful when the initial velocity in the longitudinal direction was reduced to $9\ m/s$.

2) Overtaking a low speed vehicle in front: In addition to studying just merging into another lane, we also test our planner on overtaking a slowly moving car in front by lane changing twice (second time to merge back into the original lane of the ego vehicle). In layered planning techniques, these kinds of scenarios are typically tackled by considering the obstacle to be static for a few seconds. Hence, this approach proves to be conservative. Through our experiments with our



(a) Longitudinal acceleration



(b) *matplotlib* animation

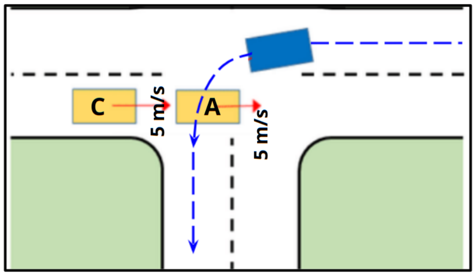
direct optimization approach, we concluded that within the comfort-optimal limits for acceleration and jerk, the planner cannot find a complete trajectory for a time horizon of $7\ s$, but was successful when we increased the horizon to $10\ s$. The differences in the longitudinal acceleration graphs between the two corridor generation techniques can be seen in Fig. 4a. Clearly, using the trapezoidal corridors generates a trajectory with much lower acceleration by utilizing the extra search space. Here, the vehicle in front is assumed to be moving with $v_s = 5\ m/s$ and the ego vehicle's initial condition is $v_s = 7.0\ m/s$. We visualize this scenario using $matplotlib$ animations as seen in Fig. 4b.

B. Qualitative Results

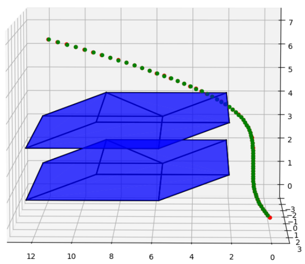
To verify that our proposed method can automatically adapt to different traffic configurations other than yielding and overtaking on straight roads, we choose to verify our planner on an unprotected left turn scenario at an intersection. As shown in Fig. 4a, we assume that there are two cars coming from the front which obstruct the ego vehicle from making a left turn without yielding to them. As seen in Fig. 4b, our planner could successfully find a trajectory while meeting all the safety and dynamic feasibility constraints for the entire time horizon of $7\ s$. Since the ego vehicle needs to yield to the cars in front, we also tested the maximum initial velocity ($v_s = 1\ m/s$) and acceleration ($a_s = 0.5\ m/s$) in the longitudinal direction for this case. If the distance between Car A and Car B is sufficient for the ego vehicle to go in between them, our planner finds the corresponding trajectory (Fig. 4c). In this scenario, the additional search space obtained by trapezoidal corridors is not used at all, as the trajectory passing through the enlarged search space can only result in a lane change which was not desired. Hence, both the trajectories obtained are the same and overlap each other, as seen in Fig. 4.

C. CommonRoad Simulations

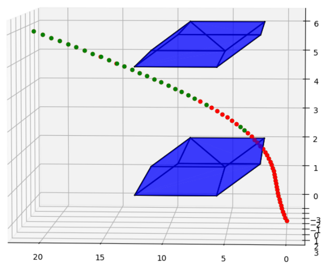
The simulations in this part are conducted using the CommonRoad [26] toolbox. It is a platform providing interactive simulated and non-interactive real traffic data. A given scenario is considered "solved" when the ego vehicle reaches the desired goal region while satisfying all the constraints. We demonstrated how the bird's-eye view simulation of the lane change scenario presented in Fig. 1 will look for our planner. The results can be seen in Fig. 5. In this case,



(a) ego vehicle is shown in blue with the corresponding reference trajectory

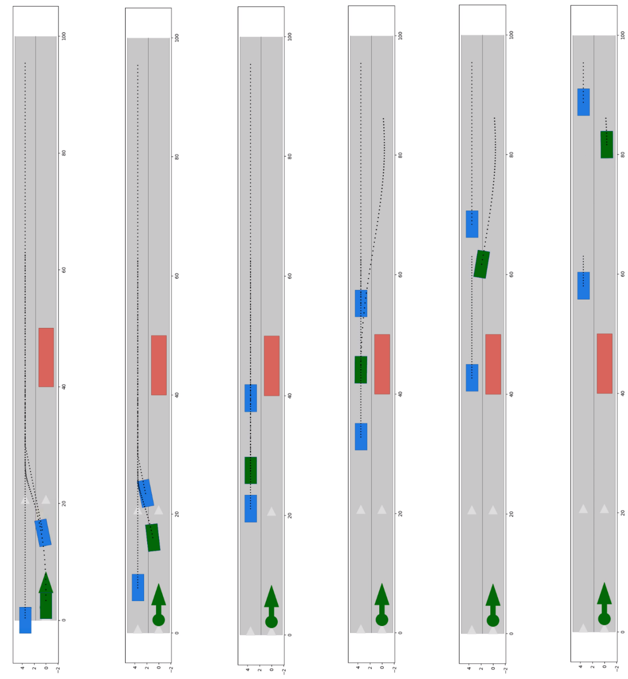


(b)



(c)

Fig. 4: Unprotected Left Turn at an Intersection- trajectories obtained from both corridor construction methods (green: trapezoidal, red: cuboidal) are identical



(a) $t = 0\ s$ (b) $t = 2\ s$ (c) $t = 4\ s$ (d) $t = 7\ s$ (e) $t = 9\ s$ (f) $t = 12\ s$

Fig. 5: CommonRoad Simulation for "Merging into another lane due to road construction" scenario. The ego vehicle is shown in green, and the other cars are shown in blue. The construction site (static obstacle) is shown in red.

replanning was only carried out once after the first horizon was about to end. CommonRoad's in-built $Route\ Planner$ is used to generate a reference trajectory for the planner. Future work involves implementation of replanning at every time step ($0.1\ s$ in this case), and testing the planner on different scenarios from the CommonRoad benchmark suite.

VII. CONCLUSION

In this paper, we investigate speed planning for autonomous vehicles. We propose a novel convexification algorithm for generating safety corridors in the $S-L-T$ space. We show that our method of trapezoidal prism-shaped corridors enlarges the solution space as compared to the existing cuboidal corridors-based method. We provide the sufficient conditions of control points in the trapezoidal corridors to provably guarantee the safety of trajectories represented by Bézier polynomials. Finally, we formulate the trajectory optimization as a QP problem. The numerical simulations show that the proposed approach is superior in terms of optimality and low failure rates. Further, we also test our planner's qualitative performance. Matplotlib and CommonRoad are used to visualize the trajectory obtained by our planner. Future work includes using a dynamic programming-based approach to generate a comfort-optimal reference trajectory in the $S-L-T$ space. We believe that such a reference trajectory will provide a better warm start to our convex optimization problem than the existing usage of piecewise functions.

APPENDIX

A. QP Formulation

This part illustrates how to formulate the Bézier polynomial optimization as a QP problem. First, we express the Bézier curve as a polynomial. Since the general equation for both S and L coordinates is the same, we only show the simplification for the polynomial for longitudinal direction $s_k(t)$:

$$s_k(t) = h_k \sum_{i=0}^{n} c_i^k b_n^i \left(\frac{t-T_k}{h_k}\right)$$
$$= h_k \sum_{i=0}^{n} p_i^k \left(\frac{t-T_k}{h_k}\right)^i = h_k f_k \left(\frac{t-T_k}{h_k}\right),$$

where $f_k(t) = \sum_{i=0}^{n} p_i^k t^i, \quad k = 0, 1, \ldots, m$ is a polynomial curve. Let $M \in \mathbb{R}^{(n+1)\times(n+1)}$ denote the transition matrix from the Bernstein basis $\left\{b_n^0(t), b_n^1(t), \ldots, b_n^n(t)\right\}$ to the monomial basis $\left\{1, t, t^2, \ldots, t^n\right\}$. Then, we have $\mathbf{c^k} = M\mathbf{p^k}$ with $\mathbf{c^k} = \left[c_0^k, \ldots, c_n^k\right]^T$ and $\mathbf{p^k} = \left[p_0^k, \ldots, p_n^k\right]^T$. According to lemma 1, it holds that $|M| > 0$ and M is invertible. Hence, if the objective function can be written as

$$J = \sum_{k=0}^{m} \left[\left(\mathbf{p^k}\right)^T Q^k \mathbf{p^k} + \mathbf{q^k p^k}\right] + \text{ const } \geq 0,$$

where Q^k is positive definite and known, then we have

$$J = \begin{bmatrix} \mathbf{c_s^0} \\ \vdots \\ \mathbf{c_s^m} \\ \mathbf{c_l^0} \\ \vdots \\ \mathbf{c_l^m} \end{bmatrix}^T \begin{bmatrix} M^{-T}Q^0M^{-1} & & 0 \\ & \ddots & \\ 0 & & M^{-T}Q^mM^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{c_s^0} \\ \vdots \\ \mathbf{c_s^m} \\ \mathbf{c_l^0} \\ \vdots \\ \mathbf{c_l^m} \end{bmatrix}$$

$$+ \begin{bmatrix} \mathbf{q^0} \\ \vdots \\ \mathbf{q^m} \end{bmatrix} \begin{bmatrix} M^{-1} & & 0 \\ & \ddots & \\ 0 & & M^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{c_s^0} \\ \vdots \\ \mathbf{c_s^m} \\ \mathbf{c_l^0} \\ \vdots \\ \mathbf{c_l^m} \end{bmatrix} + \text{ const}$$

Q_c is also a positive-definite matrix. Since the constraints are all linear in $\mathbf{c}$, the optimization problem is a QP problem.

Next we will illustrate that equation (1) holds and how to calculate Q_k and $\mathbf{q^k}$. We first calculate some terms to achieve the cost function J. To begin with, it holds that

$$\int_{T_k}^{T_{k+1}} \left(\frac{\mathrm{d}^l s(\tau)}{\mathrm{d}\tau^l}\right)^2 \mathrm{d}\tau = \int_0^{h_k} \left(\frac{\mathrm{d}^l s\left(\tau + T_k\right)}{\mathrm{d}\tau^l}\right)^2 \mathrm{d}\tau$$

$$= \int_0^{h_k} \left(\frac{\mathrm{d}^l s\left(\tau + T_k\right)}{\mathrm{d}t^l}\left(\frac{\mathrm{d}t}{\mathrm{d}\tau}\right)^l\right)^2 \mathrm{d}\tau$$

$$= \frac{1}{h_k^{2l-3}} \int_0^1 \left(\frac{\mathrm{d}^l f_k(t)}{\mathrm{d}t^l}\right)^2 \mathrm{d}t$$

As for $\int_0^1 \left(\frac{\mathrm{d}^l f_k(t)}{\mathrm{d}t^l}\right)^2 \mathrm{d}t$, it follows that

$$\int_0^1 \left(\frac{\mathrm{d}^l f_k(t)}{\mathrm{d}t^l}\right)^2 \mathrm{d}t = \int_0^1 \sum_{i\geq l, j\geq l} p_i^k p_j^k t^{i+j-2l} \,\mathrm{d}t$$

$$= \sum_{i\geq l, j\geq l} \frac{i(i-1)\cdots(i-l)j(j-1)\cdots(j-l)}{i+j+1-2l} p_i^k p_j^k$$

We also have $\int_0^1 t f_k(t)\mathrm{d}t = \sum_i \frac{1}{i+2}p_i^k, \int_0^1 f_k(t)\mathrm{d}t = \sum_i \frac{1}{i+1}p_i^k$. Suppose $J_s = \sum_{i=1}^5 w_i J_i$, the terms of J_s satify

$$J_1 = \sum_{k=0}^m \int_{T_k}^{T_{k+1}} \left(s_k(t) - a_k\left(t - T_k\right) - b_k\right)^2 \,\mathrm{d}t$$

$$= \sum_{k=0}^m \int_{T_k}^{T_{k+1}} \left[s_k(t)^2 - 2\left(a_k\left(t - T_k\right) + b_k\right) s_k(t)\right] \mathrm{d}t + \text{ const}$$

$$= \sum_{k=0}^m h_k^3 \int_0^1 f_k(t)^2 \,\mathrm{d}t - 2h_k^3 a_k \int_0^1 t f_k(t)\mathrm{d}t - 2h_k^2 b_k \int_0^1 f_k(t)\mathrm{d}t + \text{ const}$$

$$J_2 = \sum_{k=0}^m \int_{T_k}^{T_{k+1}} \dot{s}_k(t)^2 \,\mathrm{d}t - 2v_r \int_0^T \dot{s}(t)\mathrm{d}t + \text{ const}$$

$$= \sum_{k=0}^m h_k \int_0^1 \dot{f}_k(t)^2 \,\mathrm{d}t - 2v_r s(T) + \text{ const}$$

$$J_3 = \sum_{k=0}^m \frac{1}{h_k} \int_0^1 \ddot{f}_k(t)^2 \,\mathrm{d}t, \quad J_4 = \sum_{k=0}^m \frac{1}{h_k^3} \int_0^1 \dddot{f}_k(t)^2 \,\mathrm{d}t$$

$$J_5 = \left(s(T) - s^r(T)\right)^2 = s(T)^2 - 2s^r(T)s(T) + \text{ const}$$

Then we can arrive at equation 18 by replacing integral terms and using $J_s = \sum_{i=1}^5 w_i J_i$.

Similarly, we can show the same results for J_l, and formulate the QP problem.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of the NSF CMU Robotics Institute REU Site, Award 1950811 in performing this work. The authors would like to thank the RISS program for providing the opportunity to conduct this research. A special thanks to Ms. Rachel Burcin, Dr. John M. Dolan, and the entire CMU Robotics Institute community.

REFERENCES

[1] W. Ding, L. Zhang, J. Chen, and S. Shen, "Safe trajectory generation for complex urban environments using spatio-temporal semantic corridor," type: article. [Online]. Available: http://arxiv.org/abs/1906.09788

[2] T. Mercy, W. Van Loock, and G. Pipeleers, "Real-time motion planning in the presence of moving obstacles," in *2016 European Control Conference (ECC)*, pp. 1586–1591.

[3] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for bertha — a local, continuous method," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 450–457, ISSN: 1931-0587.

[4] T. Gu, J. M. Dolan, and J.-W. Lee, "Runtime-bounded tunable motion planning for autonomous driving," in *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, pp. 1301–1306. [Online]. Available: http://ieeexplore.ieee.org/document/7535558/

[5] Wenda Xu, Junqing Wei, J. M. Dolan, Huijing Zhao, and Hongbin Zha, "A real-time motion planner with trajectory optimization for autonomous vehicles," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, pp. 2061–2067. [Online]. Available: http://ieeexplore.ieee.org/document/6225063/

[6] X. Li, Z. Sun, D. Cao, Z. He, and Q. Zhu, "Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications," vol. 21, no. 2, pp. 740–753, conference Name: IEEE/ASME Transactions on Mechatronics.

[7] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong, "Baidu apollo EM motion planner," type: article. [Online]. Available: http://arxiv.org/abs/1807.08048

[8] J. Zhou, R. He, Y. Wang, S. Jiang, Z. Zhu, J. Hu, J. Miao, and Q. Luo, "Autonomous driving trajectory optimization with dual-loop iterative anchoring path smoothing and piecewise-jerk speed optimization," vol. 6, no. 2, pp. 439–446, conference Name: IEEE Robotics and Automation Letters.

[9] X. Yang and H. Li, "IA planner: Motion planning using instantaneous analysis for autonomous vehicle in the dense dynamic scenarios on highways," number: arXiv:2103.10909. [Online]. Available: http://arxiv.org/abs/2103.10909

[10] C. Liu, W. Zhan, and M. Tomizuka, "Speed profile planning in dynamic environments via temporal optimization," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 154–159.

[11] X. Qian, I. Navarro, A. de La Fortelle, and F. Moutarde, "Motion planning for urban autonomous driving using bézier curves and MPC," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 826–833, ISSN: 2153-0017.

[12] Y. Pan, Q. Lin, H. Shah, and J. M. Dolan, "Safe planning for self-driving via adaptive constrained ILQR." [Online]. Available: http://arxiv.org/abs/2003.02757

[13] Z. Zhu, E. Schmerling, and M. Pavone, "A convex optimization approach to smooth trajectories for motion planning with car-like robots." [Online]. Available: http://arxiv.org/abs/1506.01085

[14] S. M. Erlien, S. Fujita, and J. C. Gerdes, "Safe driving envelopes for shared control of ground vehicles," vol. 46, no. 21, pp. 831–836. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016384786

[15] C. Liu, C.-Y. Lin, and M. Tomizuka, "The convex feasible set algorithm for real time optimization in motion planning." [Online]. Available: http://arxiv.org/abs/1709.00627

[16] J. Li, X. Xie, J. He, Q. Lin, and J. Dolan, "Motion planning by search in derivative space and convex optimization with enlarged solution space," 07 2022.

[17] W. Zhang, P. Yadmellat, and Z. Gao, "A sufficient condition for convex hull property in general convex spatio-temporal corridors," type: article. [Online]. Available: http://arxiv.org/abs/2110.00065

[18] W. Xu and J. M. Dolan, "Speed planning in dynamic environments over a fixed path for autonomous vehicles," p. 7.

[19] M. Werling, S. Kammel, J. Ziegler, and L. Gröll, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," vol. 31, no. 3, pp. 346–359, publisher: SAGE Publications Ltd STM. [Online]. Available: https://doi.org/10.1177/0278364911423042

[20] D. González, V. Milanés, J. Pérez, and F. Nashashibi, "Speed profile generation based on quintic bézier curves for enhanced passenger comfort," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 814–819, ISSN: 2153-0017.

[21] F. Gao, W. Wu, Y. Lin, and S. Shen, "Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 344–351, ISSN: 2577-087X.

[22] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight." [Online]. Available: http://arxiv.org/abs/1907.01531

[23] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[24] Y. Zhang, H. Sun, J. Zhou, J. Hu, and J. Miao, "Optimal trajectory generation for autonomous vehicles under centripetal acceleration constraints for in-lane driving scenarios," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3619–3626.

[25] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020. [Online]. Available: https://doi.org/10.1007/s12532-020-00179-2

[26] M. Althoff, M. Koschi, and S. Manzinger, "Commonroad: Composable benchmarks for motion planning on roads," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 719–726.

Learning a Gating Function for Discrete Lossless Sparse Communication in Multi-Agent Systems

Stephane Hatgis-Kessell[1], Seth Karten[2], Dana Hughes[2], Katia Sycara[2]

Abstract— **Communication is essential for multi-agent cooperation, both for teams comprised of entirely artificial agents and teams with human agents. In addition to learning what to communicate, learning when to communicate may improve a team's performance and allow for communication budget constraints when bandwidth is limited. Recent work in addressing this decentralized sparse budget problem sacrifices performance and sample efficiency in order to constrain communication, or seeks to remove uninformative communication through manual analysis rather than reason about when it is best to communicate. We investigate how to learn a gating function that adheres to a sparse communication budget while only removing uninformative communication. By using discrete communication vectors, and taking into account previously received communications, we are able to learn a gating function that reasons about when it is best to communicate. Our proposed method is comparable to the current state of the art in terms of sample efficiency, and increases the range of lossless sparse budgets in cooperative multi-agent tasks where communication is essential for good performance.**

Index Terms— **Autonomous Agents, Multi-Robot Systems, Reinforcement Learning, Emergent Communication**

I. INTRODUCTION

There are many important tasks that require teams of robots to learn to work together. Communication between teammates can greatly aid such cooperation. When operating in partially observable environments, communication becomes necessary for multi-agent teams to successfully complete their designated task. Through emergent communication, agents may learn a communication protocol that is apt for solving a task in conjunction with learning how to successfully complete the task. This formulation presents the challenge of learning what to communicate and how to use received communication to achieve a goal. This paradigm has successfully been applied to multi-agent reinforcement learning [1]–[3]. These works, however, assume that all agents may communicate with each other continually. Such an assumption cannot hold in the real world, where agents may be subject to bandwidth constraints.

Consequently, a new challenge arises; learning when to communicate in order to adhere to a communication budget. Learning when to communicate, while simultaneously learning what to communicate and how to use such communications, presents an exceedingly difficult training challenge. Doing so in a decentralized framework requires agents to reason about what communications may actually be useful to their teammates. Previous methods that attempt to address this decentralized sparse budget problem trade off communication sparsity with task performance, suffer from high variance and instability while training [4]–[6], or require a manual analysis of what communications are important [7].

In this work we seek to restrain communication without loss in task performance by learning a decentralized gating function that both removes uninformative communication and reasons about when it is best to communicate.

We refer to communications that can be removed without any loss in performance as null communications. These communications come in many forms. If an agent does not have any meaningful information to communicate, its outputted communication may be akin to random noise. On the other hand, communication may be generally informative but not necessary at a given time. Our goal is to remove such communications through gating.

Our first contribution is utilizing a vector quantized variational autoencoder (VQ-VAE) [8] to discretize communication. Discrete communications are more interpretable by humans [9], and allow us the control the range of possible communications. The VQ-VAE has been shown to produce more meaningful communications than other methods for discretizing communications in multi-agent reinforcement learning [9]. This technique may lead to better human-agent teaming performance in future work.

Our second contribution is a learned, decentralized gating function that decides whether each agent should communicate or not. Our proposed gating function takes into account what an agent already knows and uses this to decided if the proposed communication may be beneficial to the agents teammates. Through the proposed gating function we decrease the number of null communications emitted by all agents, and increase the range of sparse communication budgets for which there is no loss in task performance when compared to prior art.

II. BACKGROUND

A. Reinforcement Learning

Reinforcement learning concerns learning what to do in order to maximize a reward signal. The reward signal defines what a learning agents goal is, and in maximizing this reward the agent will successfully have completed its designated task. Reinforcement learning can be characterized as addressing two challenges: trial-and-error search, where the learning agent must try out different actions and observe their affects, and delayed reward, where an action taken may decide the reward incurred at a later time.

A policy defines how the learning agent behaves in a given state by mapping the state to an action that the agent

[1]University of Texas at Austin
[2]Robotics Institute, Carnegie Mellon University

will execute. A value function specifies the expected future reward from any given state, where a discount may be applied at each step into the future to lessen the affect of future rewards. Both the policy and value function must by learned by the agent, however, the former may be derived from the latter and vice versa. In learning a policy and value function, a reinforcement learning agent learns how to behave so as to complete its sole objective: maximizing the reward signal.

B. Multi-Agent Reinforcement Learning

This work falls within the bounds of cooperative multi-agent reinforcement learning (MARL), where teams of agents must learn to work together in order to complete a task. Like much of the prior art in this area [6] [4] [2], we parameterize each agents policy as a deep neural network and use a policy gradient update to optimize for the task reward. Specifically, this work uses REINFORCE [10] to update each agents parameters and estimate state values, however, other policy gradient algorithms may be used as a substitution. We seek to extend the capabilities of cooperative agents by enabling them with communication.

C. Variational Autoencoders

A variational autoencoder (VAE) seeks to encode an input, x, into a latent variable, z, which may then be decoded back into the original input. The encoder network learns the parameters, θ for the distribution $q_\theta(z|x)$, which the latent variable z may then be sampled from. Note that z cannot be directly sampled from this distribution, but is instead sampled using the reparameterization trick [11]. The prior, $p(z)$ is assumed to be $N(0, 1)$. The decoding distribution with parameter ϕ, $p_\phi(x|z)$, is also learned so as to reconstruct the original input from z. In order to learn the parameters for $q_\theta(z|x)$ and $p_\phi(x|z)$, the following loss is minimized:

$$loss(\theta, \phi) = \mathbb{E}_{z\sim q_\theta(z|x)}[logp_\phi(x|z) + KL(q_\theta(z|x)||p(z))]$$

The first term of the loss above measures how well x can be reconstructed from z. The second term of the loss measures how much information is lost when using q to approximate p. We use a variation of the variational autoencoder, the VQ-VAE, to generate discrete communication vectors that represent each agents observation.

III. RELATED WORK

A. Emergent Communication

Prior work has shown that communication can significantly increase team performance with regards to multi-agent reinforcement learning. In partially observable environments communication may be essential to achieving a goal. Learning communication vectors while simultaneously learning to complete a task has proven to be successful in a variety of partially observable problems [4]. Some approaches focus on learning continuous communication vectors [4] [5] [6]. Inspired by human communication, other approaches focus on learning discrete communication vectors [7] [1] [12], which is the approach that we take in this paper. Prior work has learned discrete communication protocols by representing communications as one-hot vectors [12]. In doing so, however, all generated communications are orthogonal and of equal distance to each other. This greatly reduces the effectiveness of such communications. IMGS-MAC [7] discretizes continuous representations via ProtoNet, a decision theoretic framework that builds off techniques from the natural language processing community. Instead, we utilize the VQ-VAE framework which has been shown to produce more meaningful communications than discrete prototypes [9].

B. Decentralized Communication

In decentralized communication, communication protocols must be learned without a centralized scheduler. Each agent must individually communicate to its teammates. In contrast to centralized communication this poses a more realistic and robust solution, but significant new training challenges. To alleviate some of these problems we adopt the centralized training decentralized execution paradigm, where all agents are trained in a centralized fashion to increase stability and performance, but, at evaluation time, all agents are decentralized and operate independently from each other. This is the same approach taken by IMGS-MAC [7] and IC3Net [4].

C. Sparse Budget Problem

Bandwidth limitations pose a serious problem to communicating agents, where much of the prior art has assumed that agents are able to engage in continual communication. To address this, some previous work has focused on adhering to communication bandwidth constraints by introducing an information bottleneck, or by improving communication compression. Doing so, however, may decrease the information present in communication vectors. For many domains, it is clear when communication is necessary and when it is not, so naturally learning this notion should be desired.

Learning a communication gating function [4] [6] [7], which decides when an agent should communicate, has shown promising results in addressing the sparse budget problem, and allows for decentralized execution. Prior methods, however, suffer from high variance and instability while training and are still unable to remove uninformative communications. Methods that utilize a communication gating functions are most similar to ours, and we hope to extend and improve them. Of these methods our paper focuses on comparing our model against IC3Net [4] and IMGS-MAC [7].

[13] learns a globalized gating function which is successfully able to reduce communication without a loss in performance, however, this model requires centralized execution which we avoid.

IMGS-MAC successfully removes all uninformative communication, and is thus capable of reducing the communication frequency without incurring a loss in performance. This methodology, however, requires manually identifying uninformative communication in the first place, and then using a table lookup function to remove them in later trials. We position our work as attempting to automate this process through a gating function, where recognizing and removing such uninformative communications should be learned by the agent itself.

IV. Problem Setup

When learning emergent communication for multi-agent reinforcement learning, we formulate the problem as a decomposed, partially observable Markov decision process with communication (Dec-POMDP-Comm). Each agent is only able to partially observe the environment, so agents must communicate their observations to each other in order to perform well. Agent i does this at time step t by transmitting the communication vector c_t^i, which encodes information about agent i's observation at time step t. Also at time step t, agent i receives communications $c_t^j, ..., c_t^n$ from all other agents.

Formally, we define this problem with the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{C}, \mathcal{T}, \mathcal{R}, \mathcal{O}, \omega, \gamma)$. $\mathcal{S}$ is the set of all states and $\mathcal{A}$ is the set of all actions for agents $1..n$, including task specific actions and the action of whether to communicate or not. $\mathcal{C}$ is the set of all communications for all agents, and $\mathcal{T}$ specifies the joint environment transition dynamics such that $\mathcal{T} : \mathcal{S} \times \mathcal{A}_i, ..., \mathcal{A}_n \rightarrow \mathcal{S}$. The set of all partially observable observations is defined by ω, and the mapping of the state and joint actions to each agents observation is given by $\mathcal{O}$ such that $\mathcal{O} : \mathcal{C}_i, ..., \mathcal{C}_n \rightarrow \omega$. $\mathcal{R}$ is the reward function, and γ is the discount factor.

Our goal is to learn how to behave, what to communicate, and when to communicate. Optimizing for these objectives simultaneously presents a difficult training challenge, and in practice can be unstable and cumbersome. Because of this we adopt the training paradigm of [7] by training our model in two stages. The model is first trained to learn what to communicate and how to behave, assuming no bandwidth constraints (ie: a communication budget of 1). Next, we apply fine-tuning to train the gating function for any budget. This has been shown to reduce variance and the amount of data needed.

Our training objective is illustrated below, which is built off the objective introduced by IMGS-MAC [7].

$$\max_{\pi \rightarrow \mathcal{S} \times \mathcal{C}} \mathbb{E}[\sum_{t \in \mathcal{T}} \sum_{i \in N} (\gamma R(s,a) + \lambda_1 U(x_t^i, c_t^i) - \lambda_2 ||c_{AVG}^i - b||_2^2)]$$

By maximizing $\gamma R(s, a)$ for each agent, we are ensuring that each agent learns to maximize reward and thus how to behave. The term $\lambda_1 U(x_t^i, c_t^i)$ encourages learning a good communication protocol that is grounded in the ground truth state space, and therefore what to communicate. Finally, maximizing $\lambda_2 ||c_{AVG}^i - b||_2^2)$ encourages each agent to adhere to communication budget b and in turn learn when to communicate. More detail is described in section V.C.

V. Methodology

We now introduce the methodology of our paper. First, we describe the general architecture for sending communications and using received communications to generation actions. Here we introduce our first core contribution, the VQ-VAE to descritize communications. Next we detail our second core contribution, the learned gating function, which decides whether an agent should communicate or not. Finally, we describe the training objective that we seek to realize.

A. Communication Architecture

We build our architecture off of IC3Net [4] and IMGS-MAC [7]. The full details are shown in figure 1. First, each agents observation x^i, is provided to an LSTM and linear layer to produce an encoded observation, $\hat{x_i}$.

Next, this encoded observation is discretized into a communication vector using a VQ-VAE [8] model. As in a standard VAE, the latent representation c is sampled from $q(c|\hat{x})$ but here q is a categorical distribution. The VQ-VAE is parameterized with K embedding vectors, c, in the latent space. Each of these embedding vectors is a unique communication that may be sent. The encoded observation, $\hat{x}$, is mapped to the closest embedding vector, c, and this is used as the agents communication. In doing so, we are able to map the observation embedding, $\hat{x}$, to one of K possible communications resulting in discrete communication while also maintaining a distribution $q(c|\hat{x})$ for all sent communications. The variational aspect of this approach may allow for later analysis of each communications entropy during gating, and the discrete aspect moves the emergent communication protocol closer to human interpretability.

The resulting communication vector is passed through a gating function, which returns a zero-vector if the agent has decided not to communicate and the original communication,c_i, otherwise. This gating function is described in detail in the next section.

Finally, all communications are sent out, and all agents receive the communications from their teammates. These received communications are concatenated with the agents own observation embedding, passing through a linear layer, and then serve as input to the action and value heads. These heads output the agents intended next action and the corresponding predicted value.

The forward pass is entirely decentralized. When training, a centralized decoder is utilized to reconstruct the original state from all agents emitted communication vectors. REINFORCE is also used to compute the policy gradient update from the outputted actions. Each agents observation encoder and decoder is then updated accordingly.

B. Gating

Intuitively, we would like each agent to reason about whether its proposed communication will be useful given what the agent already knows. To this end, each agent has its own decentralized gating function.

First the agents proposed communication, c_t^n, as well as the last received communications from all other agents, $c_{t-i}^1, ..., c_{t-j}^{n-1}$, serve as input to a single linear layer. The log-Softmax of the output of this layer, z', is then used to sample from a Gumbel-Softmax distribution [14] to produce

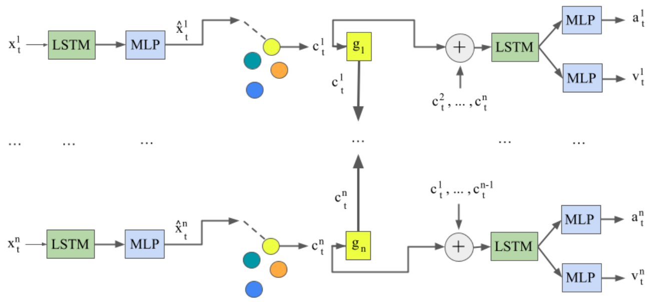


Fig. 1. Overview of the complete communication and behaviour architecture.

the gating action, w. Sampling the gating action from a Gumbel-Softmax distribution allows us to sample from a categorical distribution without breaking the gradient signal. The Hadamard product between w and c_t^n is then outputted by the function.

The scheme is illustrated in figure 2. This sampled gating action is used by REINFORCE to update the gating function parameters. The output of the gating function is then the Hadamard product between the gating action and the inputted communication vector, which results in either the inputted communication vector if the agent decides to communicate ($w_i = 1$) or a zero vector if the agent has decides not to communicate ($w_i = 0$).

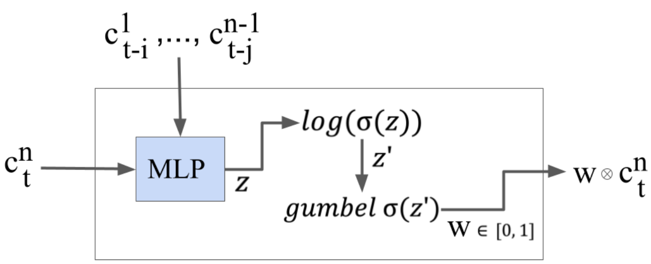


Fig. 2. Overview of the gating architecture.

C. Objective

objective should go in problem setup I disagree with this only because there is a lot of technical description that rests on the methedology section needed to fully understand the objective. I do, however, describe the objective in plain English in the problem setup. Plain english can be repeated. The formalization should be in the problem setup section. Papers often rephrase and repeat info since they are not always read in order. If I were looking to see what formal problem you are solving, and that info is not in the problem setup, that is a red flag. fix.

We build on the objective introduced by IMGS-MAC [7], where we optimize the total expected reward for all agents while simultaneously learning a good communication protocol and communication policy. This objective is shown below.

$$\max_{\pi \to \mathcal{S}\times\mathcal{C}} \mathbb{E}[\textstyle\sum_{t\in\mathcal{T}}\sum_{i\in N}(\gamma R(s,a)+\lambda_1 U(x_t^i,c_t^i)-\lambda_2||c^i_{AVG}-b||_2^2)]$$

Where $\lambda_1 U(s_t, c_t^i)$ measures the quality of the generated communication vector for agent i such that

$$U(s_t,c_t^i) = -log(p(s_t|c_t^i)) - ||sg(q(z|s_t)) - c_t^i||_2^2 + \beta||q(z|s_t) - sg(c_t^i)||_2^2$$

The first term is the reconstruction loss between the communication vector c_t^i and the original state s_t. The second term aims to move the communication vector, c_t^i, closer to the encoder output $q(z|s_t)$. We apply a stop gradient, denoted as $sg(.)$, to prevent the gradient here from updating the encoders distribution. Finally, the third term encourages the encoder to commit to a communication vector.

The last term of the objective, $\lambda_2||c^i_{AVG} - b||_2^2)]$, measures the difference between the average rate of communication and the budget, thereby enforcing the bandwidth constraints. The parameters λ_1 and λ_2 are tuned at training time.

VI. EXPERIMENTS

A. Setup

We train and evaluate our model in a blind traffic junction setting, where communication is essential for good performance. In this setting, multiple agents must navigate to a predefined goal without colliding. All agents can only partially observe the environment, however, and have no knowledge of where the other agents are. Therefore, agents must communicate their positions to their teammates in order to avoid collisions and successfully navigate intersections.

The traffic junction environment is comprised of discrete cells with specified entry and exit cells for each agent. Agents are spawned at entry cells randomly, and the rate at which they spawn as well as the number of cells in the environment is defined by each difficulty level. While in the environment, the agent may only execute one of two actions: start or stop. If agents collide then the episode terminates in failure.

In the easy level of traffic junction, there are 7 cells that form a two-way intersection, and a maximum of 5 agents at any given time. For easy traffic junction the model is trained with 1 processes using a batch size of 500. Results were averaged over 10 gradient updates for each back propagation step. We used an RMSProp optimizer with a learning rate of 0.001. Unless stated otherwise, the model is always evaluated across 5,000 episodes. Each episode has a length of 20 steps, and we refer to an episode as being successful if there are no collisions.

Our training paradigm begins with training the model with non-sparse communication (b=1). Results for this first stage are shown below in table 1, where a model converges to a solution when it consistently achieves a success rate of 97% or above. These results indicate that our method is comparable in terms of sample efficiency and performance to IMGS-MAC, and is significantly more performant than IC3Net.

Easy Traffic Junction	
Model	Number of Epochs Until Convergence
IMGS-MAC	495 ± 2
IC3Net	> 600
Ours	500 ± 5

TABLE I

NUMBER OF EPOCHS UNTIL EACH MODEL CONSISTENTLY ACHIEVES $\geq 97\%$ SUCCESS WHEN TRAINED WITHOUT SPARSE COMMUNICATION ($b = 1$).

B. Sparsity Analysis

We now seek to minimize the amount of communication without loss in performance. This enacts the second stage of the training paradigm, where we use the model trained with non-sparse communication and update its gating functions parameters to minimize the sparse communication penalty, $\lambda_2 ||c^i_{AVG} - b||^2_2$, while maximizing the sum of agents rewards.

We perform this procedure for a range of budgets from $b = 0.9$ to $b = 0.1$, and for each seed select b^* such that it is the minimum budget that has not incurred a loss in performance at evaluation time. We define a loss in performance for a given seed more formally below, where $\mu_{success,b=x}$ is the mean success at evaluation time for a model trained with budget x, $\mu_{success,b=1}$ is the mean success at evaluation time for the model trained without sparse communication, and $SE_{success,b=1}$ is the standard error of success for the model trained without sparse communication.

$\mu_{success,b=x} < \mu_{success,b=1} - 2SE_{success,b=1}$

Our model is able to reduce the minimum lossless budget when compared to IMGS-MAC, and does so without any manual analysis of learned communication protocol. That being said, our model is higher in variance and less stable. This can be seen in table 2.

Easy Traffic Junction	
Model	Min Budget b^*
IMGS-MAC	0.815 ± 0.00469
Ours	0.640 ± 0.37736

TABLE II

MINIMUM SPARSE BUDGET b^* WITH LOSSLESS PERFORMANCE. OBSERVE THAT OUR MODEL IS ABLE TO INCREASE THE RANGE OF LOSSLESS SPARSE BUDGETS. THE IMGS-MAC MIN BUDGET USES YOUR METHODOLOGY RATHER THAN THE IMGS-MAC METHODOLOGY. NOT SURE IF THIS A FAIR COMPARISON. YOU NEED TO LIST THE NULL COMMUNICATION ANALYSIS METHOD TOO TO HAVE A FAIR COMPARISON.

Ideally, when gating communication in order to increase sparsity, the model should be learning to identify and remove null communications. To evaluate this claim, we seek to identify outputted null communications. The process to do this is as follows.

For each of the K possible communications, $c_1, ..., c_K$, we evaluate the trained model, allowing the agents to emit all communications except for the designated communication c_i. If there is no loss in performance when omitting this communication it is considered a null communication. We then count the number of null communications emitted during evaluation time when all K possible communications are permitted.

Following this analysis our results in table 3 indicate that the model is somewhat successful at removing null communications. While we are able to reduce the number of null communications emitted by each agent, we do not totally eliminate them.

Easy Traffic Junction	
% null communications when $b = 1$	% null communications when $b = b^*$
0.459 ± 0.27234	0.275 ± 0.26838

TABLE III

MINIMUM SPARSE BUDGET b^* WITH LOSSLESS PERFORMANCE. OBSERVE THAT OUR MODEL IS ABLE TO INCREASE THE RANGE OF LOSSLESS SPARSE BUDGETS.

VII. CONCLUSION AND FUTURE WORK

In this work we have built upon prior art by proposing a modified gating function and a methodology for discretizing communication using the VQ-VAE framework. When compared to prior work, our model reduces the range of lossless communication budgets in a partially observable, multi-agent environment where good communication is paramount. While our proposed methodology is comparable to prior work in terms of sample efficiency, significant improvements can be made in future work by exploring different communication architectures. We also hope to create a more informed gating function that looks at the relative information between proposed communications and previously received communications, further addressing the decentralized sparse budget problem. Finally, a natural extension to the problem of learning when to communicate is learning with whom to communicate, which may further increase the range of lossless communication budgets. With regard to the broader impact of this work, learning good discrete communication protocols and learning when such communications are necessary may greatly increase the capabilities of human-agent teams, where humans are subject to cognitive load constraints. Such work may also benefit teams comprised entirely of agents, where sending communications may be risky due to possible security breaches or general bandwidth constraints.

ACKNOWLEDGMENT

I would like to thank Rachel Burcin and John Dolan for hosting the Robotics Institute Summer Scholars program, giving me the opportunity to work on this project with such amazing people. Funding for this work was provided by the NSF REU initiative.

References

[1] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," *Advances in neural information processing systems*, vol. 29, 2016.

[2] S. Sukhbaatar, R. Fergus, *et al.*, "Learning multiagent communication with backpropagation," *Advances in neural information processing systems*, vol. 29, 2016.

[3] B. Freed, G. Sartoretti, and H. Choset, "Simultaneous policy and discrete communication learning for multi-agent cooperation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2498–2505, 2020.

[4] A. Singh, T. Jain, and S. Sukhbaatar, "Learning when to communicate at scale in multiagent cooperative and competitive tasks," *arXiv preprint arXiv:1812.09755*, 2018.

[5] Z. Ding, T. Huang, and Z. Lu, "Learning individually inferred communication for multi-agent cooperation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 22 069–22 079, 2020.

[6] V. K. Vijay, H. Sheikh, S. Majumdar, and M. Phielipp, "Minimizing communication while maximizing performance in multi-agent reinforcement learning," *arXiv preprint arXiv:2106.08482*, 2021.

[7] S. Karten, M. Tucker, S. Kailas, D. Hughes, and K. Sycara, "Towards true lossless stable sparse communication in multi-agent systems," 2022.

[8] A. Van Den Oord, O. Vinyals, *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[9] M. Tucker, J. Shah, R. Levy, and N. Zaslavsky, "Towards human-agent communication via the information bottleneck principle," *arXiv preprint arXiv:2207.00088*, 2022.

[10] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in neural information processing systems*, vol. 12, 1999.

[11] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[12] S. Li, Y. Zhou, R. Allen, and M. J. Kochenderfer, "Learning emergent discrete message communication for cooperative reinforcement learning," *arXiv preprint arXiv:2102.12550*, 2021.

[13] H. Mao, Z. Zhang, Z. Xiao, Z. Gong, and Y. Ni, "Learning agent communication under limited bandwidth by message pruning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5142–5149.

[14] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[15] S. Karten, M. Tucker, H. Li, S. Kailas, M. Lewis, and K. Sycara, "Interpretable learned emergent communication for human-agent teams," 2022.

The microDelta: A 3D-printed micro-scale Delta robot

Steven Man[1], Sukjun Kim[2], and Sarah Bergbreiter[2]

Abstract— **Delta robots are suitable for assembly, mobility, and stability tasks such as 3D printing and pick-and-place applications for their precision and speed. When integrated with Two-Photon Polymerization (TPP) fabrication techniques, we are able to design and fabricate compliant Delta mechanisms on the micro-scale, which provides 3-degrees of freedom (DOF) motion. Although some microelectromechanical systems (MEMS) can operate out of plane and have more than 3-DOF, but the design and fabrication processes to achieve such results involve complex procedures like layer bonding and alignment since most semiconductor processes only create 2D features. We use 3D printed torsional comb-drive actuators to provide revolute input into our Delta mechanism. The manufacturing process of the micro-robot uses gold sputtering to functionalize the polymer structures and isotropic etching to release structures from the underlying substrate. Using this process, we demonstrate a 3D printed micro-robotic system that includes mechanisms, and actuation for the first time.**

Index Terms— **3D-Printing, Compliant Joints and Mechanisms, Mechanism Design, Micro/Nano Robots, Microactuator Parallel Robots, Two-Photon Polymerization (TPP)**

I. INTRODUCTION

3D printing technology allows for design and fabrication of micro-robots while offering minimal post-processing complexity [1]. Micro-robots are important for a variety of manipulation tasks at small scales with high precision.

The milliDelta [2] pushed the limits of smart composite microstructures (SCM) technology and demonstrated its ability to manufacture millimeter scale robotics with high-precision and 3-DOF. However, the need for manual assembly makes the process of creating sub-millimeter robots with this method difficult and complex. On the other end of the spectrum, the MEMS community has long been able to make sub-micron, 2D-extruded features on micro-scale robots [3] using CMOS (complementary metal-oxide semiconductor) and MEMS techniques in the cleanroom; still, it is difficult to create 3-DOF motion due to the planar nature of most cleanroom fabrication processes like photo-lithography, thin-film deposition, and etching.

We introduce microDelta, a 3D-printed micro-scale Delta robot. It uses actuators built upon the torsional comb-drive actuator (TCDA) [4] manufactured in the same method. Pin joints on traditional Delta mechanism have been replaced with flexural joints so the entire robot can be printed at once without assembly as well as being free from backlash. The 3D-printing process is followed by dry-etching of the substrate and metal sputtering to rapidly produce usable robots.

[1] Steven Man is with the College of Engineering, University of California Santa Barbara, CA, USA

[2] Sukjun Kim and Sarah Bergbreiter are with the Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA

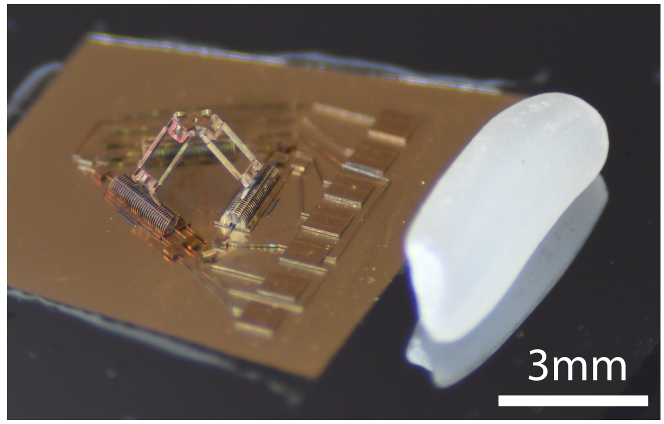


Fig. 1. The microDelta next to a grain of rice. It has a circular footprint of less than 3 mm in diameter and a resting platform height of 1.3 mm. When actuated, the platform can be lowered to 900 µm

II. DESIGN

The microDelta robot is designed using 3D computer-aided design (CAD) software (SolidWorks) where the main design considerations stem from the desired structural requirements such as mechanism compliance as well as electrical characteristics like actuator torque at a given voltage. An isometric view is presented in Fig. 2 where the flexible joints and actuator parameters are labeled.

A. Torsional Comb-Drive Actuator

The actuator consists of round combs with torsional springs attached symmetrically. The overall design principles are similar to the TCDA [4], with minor changes to adapt to the need to rotate in both clockwise and counterclockwise directions (Fig. 2), doubling the maximum angular displacement. With removal of the center beam, individual comb fingers could collapse during fabrication so three spars are added on top to prevent such failure. The torque that the actuator can generate depends on the number of comb fingers, the area between overlapping fingers, and the gap between two fingers; their relationship is outlined in section III-A. By varying the geometry of comb fingers or torsional springs, we control the actuator's voltage-displacement relationship.

B. Flexure-Based Delta Mechanism

The 3-DOF Delta mechanism consists of three kinematic loops between two parallel plates. Each loop contains a base arm, pinned at the base, connecting to a parallelogram

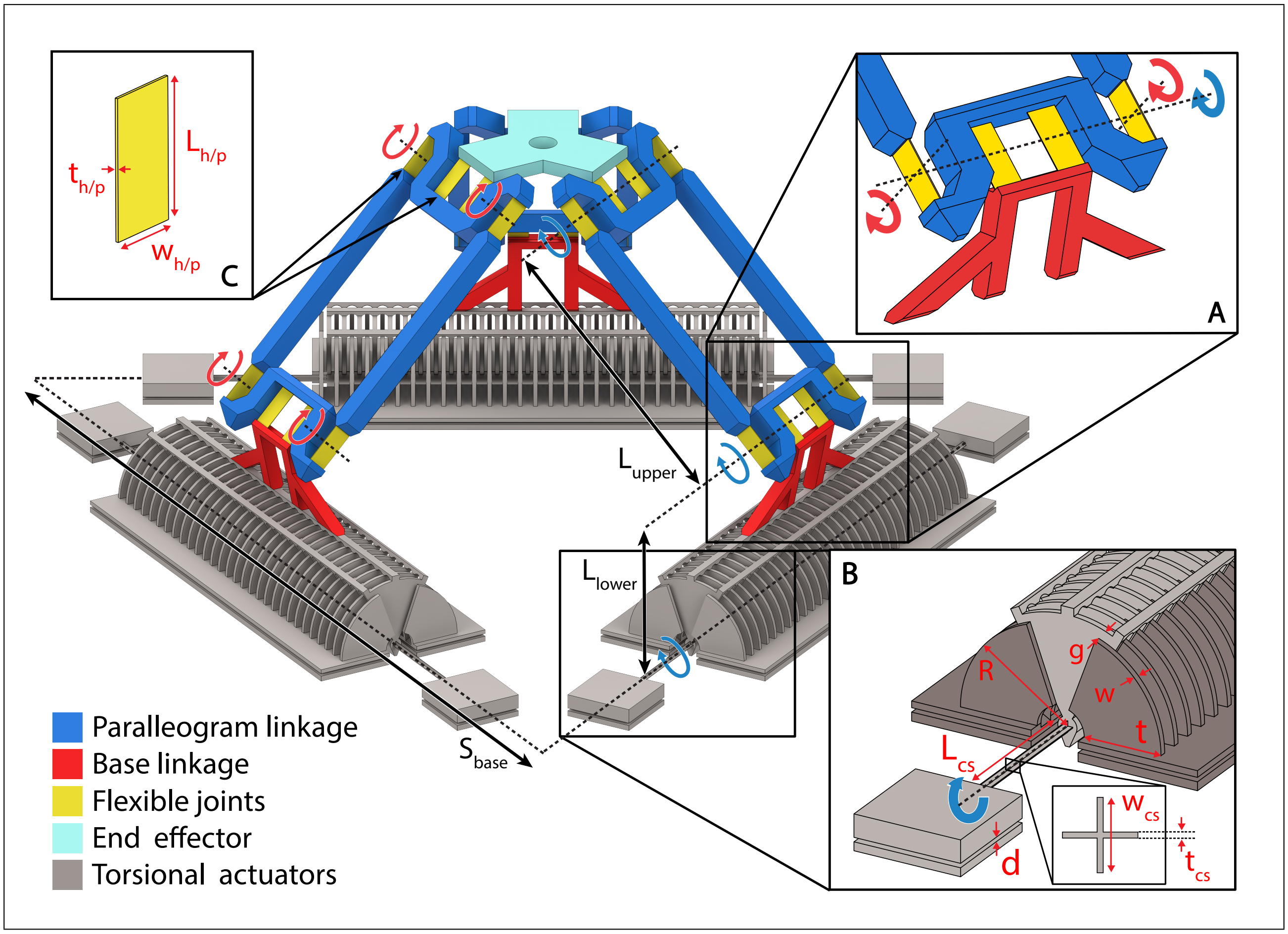


Fig. 2. The microDelta: a 3D-printed micro-scale Delta robot. This robot is 3D-printed using two-photo polymerization and driven by 3D comb-drive actuators. (A) The design of the torsional flexural joints aligns the centers of parallelogram joints (rotates along the red arrows) with the centers of the hinge joints (rotates along the blue arrow). (B) The critical dimensions in designing a torsional comb-drive actuator. (C) Geometric dimensions of hinge and parallelogram flexures. All labeled dimensions are listed in Table I, for flexure geometries, subscript *h* represents hinge joints (connects between cyan, blue, and red-label components) while *p* represents parallelogram joints (connects blue-label components)

linkage through a universal joint, connected to the end-effector plate through another universal joint (Fig. 2). Like the milliDelta, our design replaced the robot's universal joints with revolute flexure joints. The pin joint on the stationary plate is replaced with actuators such that the center of the torsional spring is coaxial with the center of the supposed pin joint location. For the milliDelta in [2], center axes of the revolute flexure hinge joints do not intersect the parallelogram joints; the microDelta resolves this misalignment by inserting an offset in the link in contact with both of the flexure joints (See Fig. 2A), which illustrates that 3D-printing can overcome design constraints posed by SCM. Certain design considerations that must be examined when printing with TPP is outlines in section IV.

All design parameters are for both actuator, flexural joints, and mechanism linkage are listed in Table I.

III. MODELING

A. Actuator Modeling

The TCDA is driven by the electrostatic force between oppositely charged objects. The analytical relationship between torque (T) generated by the actuator and applied voltage (V) as derived in [4] is:

$$T = \frac{1}{2}\, N_f\, \varepsilon_0\, \varepsilon_r \left(\frac{2\, R\, t - t^2}{g}\right) V^2 \tag{1}$$

where N_f is the number of moving comb fingers, ε_0 is the vacuum permittivity, ε_r is the relative permittivity of air, R is the outer radius of comb fingers, t is the thickness of the comb fingers (difference in radius between the inner and outer arc of the finger), and g is the gap between two adjacent comb fingers (see Fig. 2B).

B. Delta Mechanism Kinematics and Quasi-statics

The kinematics of Delta mechanisms have been widely studied since its conception. Given conventional Delta robots with pin joints, forward and inverse position kinematic solutions from [5] would suffice in providing basic insight on how to drive each revolute actuation input at the base. However, due to stiffness inherent to microDelta's revolute flexure joints, and the actuator's open-loop configuration, driving any of the three actuators with some non-zero torque would result in internal resistance torque, as well as displacement, at all

TABLE I

The microDelta's design parameters. Electrostatic actuator and Delta mechanism's geometric dimensions.

The microDelta's Parameters			
Actuator		Mechanism	
N_f	30	S_{base}	3606 μm
R	235 μm	L_{lower}	579 μm
t	180 μm	L_{upper}	1020 μm
w	15 μm	L_h	150 μm
g	10 μm	w_h	75 μm
d	10 μm	t_h	3.95 μm
L_{CS}	400 μm	L_p	150 μm
w_{CS}	24.2 μm	w_p	75 μm
t_{CS}	2.71 μm	t_p	3.79 μm

three actuators, the coupled behavior will be analyzed in a quasi-static state where all joint stiffness must be taken into account. In this case, flexure joints will be modeled as linear torsional springs as a pseudo-rigid-body model of a small length flexural pivot [6]:

$$T_{joint} = k_{joint}\Delta\theta \tag{2}$$

where joint torque (T_{joint}) generated through spring deflection ($\Delta\theta$) with stiffness ($k_{joint} = \frac{EI}{L}$, where E, I, and L are the Young's modulus, second moment of area, and length of the flexural joints).

We used MATLAB's Simscape Multibody software to model the joints. Geometric parameters of the mechanism such as linkage length and joint angles are provided by the 3D CAD model. 1-DOF revolute joints were positioned at each fixture, each joint's stiffness were determined by its geometry, as well as verified during actuator and mechanism characterization procedures outlined in section V-B. Kinematic loop constraints are enforced at the end-effector platform. To actuate the microDelta, we used the inverse kinematic equations with the end-effector position as input and obtained the required angular position of the three actuators. We then input actuator deflections as constants into the Multibody simulation and recorded actuator torques as outputs. Actuator input voltages were then converted from actuator torques using Eq. (1).

IV. FABRICATION

The microDelta is printed with Nanoscribe Photonic Professional GT+ using Dip-In Laser Lithography (DiLL) and a negative photoresist (IPS, Nanoscribe) on a silicon substrate (Fig. 3A). After the photoresist was developed by propylene glycol methyl ether acetate (PGMEA, Sigma-Aldrich) the sample was cleaned in isopropyl alcohol (IPA, VWR) and air-dried in room temperature. The silicon substrate was then dry-etched with XeF_2 (SPTS Xactix, Xetch) to release the moving combs and the torsional springs (Fig. 3B). Finally, 50 nm of gold was sputter deposited over the actuator to make the structure electrically conductive (Perkin Elmer, 2400-8L), see Fig. 3C.

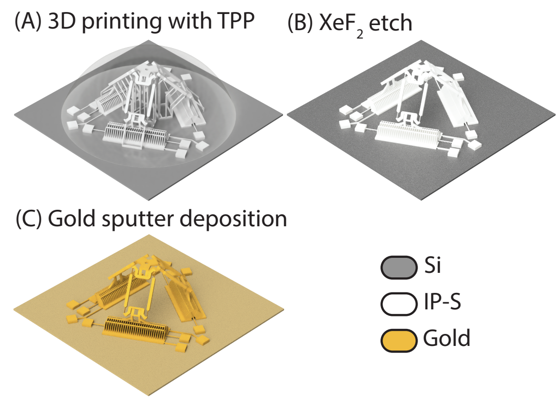


Fig. 3. Fabrication process of microDelta. (A) Desired actuator and mechanism structures are printed on a silicon substrate using TPP. (B) The silicon substrate is etched with XeF_2 to release hanging structures after developing the print and removing supports. (C) Gold is sputtered over the entire 3D structure.

When designing for fabrication, supports were inserted in the design and manually removed afterwards to prevent rigid bodies between compliant joints from drifting apart during printing and stage movement. Typically supports are designed to interface the surfaces of the supported part through a 2 μm diameter circle to allow removal with minimal damage while capable of resisting drifting. Another design consideration is to ensure the structural stability of comb fingers and its resistance to stiction caused by surface tension of IPA during development and drying, when fingers have high aspect ratio cross-sections, they tend to collapse to neighboring fingers on either side and increasing the widths w and gap g (Fig. 2B) would prevent such failure modes.

Actuators used in microDelta differs from previous TCDA [4] in that gold was sputtered over the polymer instead of aluminum. Gold-deposition provides better side-wall coverage when compared to aluminum, which succumbs to shadow-masking and are unable to cover features under other overhanging structures. Although gold is more easily deposited across the surfaces of the 3D-printed structures, certain features with interior notches (Fig. 2B) take advantage of the coverage-limiting techniques used in [7] to insulate features from each other and the substrate, achieving the required electrical isolate between voltage terminals. The notch has a vertical gap d of 10 μm and the notch offsets inward for 25 μm.

V. RESULTS

A. Experimental Setup

To actuate the microDelta, an Arduino controls a digital-analog converter (DAC) chip (LTC2637, Analog Devices) whose output voltages are then amplified by an operational amplifier (HV265, Microchip). The actuator terminals are probed under the probe station (S-250-6, Signatone) with a custom made multi-probe needle manipulator (see Fig. 4), where the amplified voltage actuates the TCDA. Using a

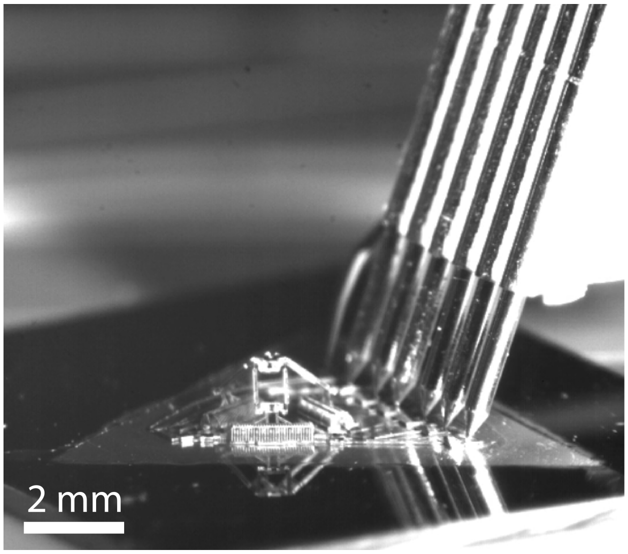


Fig. 4. Isometric view of the microDelta with multi-needle setup. The custom multi-needle manipulator connects the voltage signal to the actuator via probing the electrical contact pads. The camera sits on the microscope objective (not in view) directly above the sample

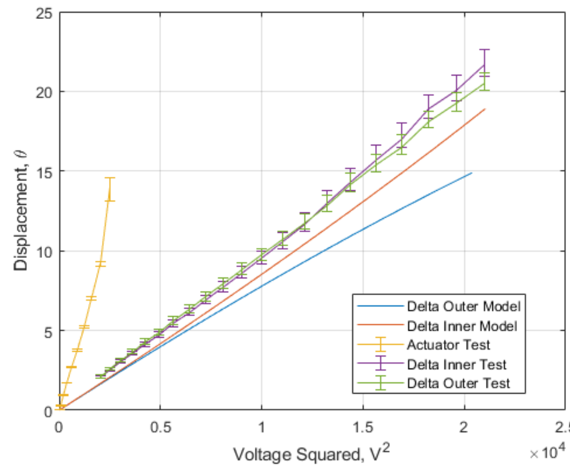


Fig. 5. Actuator and Delta mechanism stiffness characterization. Delta inner and outer model curves were obtained using the Simscape Multibody model while actuator, Delta inner, and Delta outer tests were performed using voltage sweeps.

DSLR camera (D850, Nikon), the motion of the actuator platform was captured and analyzed with motion analysis software (TEMA T2020, Image Systems). Desired paths are discretized and corresponding torques at each actuators are obtained using MATLAB as mentioned in section III-B.

B. Stiffness Characterization and Calibration

The flexural joint stiffnesses of the microDelta were initially modeled using geometric and material properties during design, assuming ideal fabrication results. Before further testing, the actuator and Delta mechanism stiffness were characterized by recording angular displacement at the base when a voltage sweep is applied to a single actuator (see Fig. 5). For the actuator only case, a standalone TCDA was tested and the cross-shaped torsional spring (Fig. 2B) of the actuator was measured to be 7.46 nNm/rad. We then tested the inner and outer actuators of a single TCDA that is a part of the microDelta. The result illustrates that the the actuator experiences inner and outer resistance symmetrically and linearly for displacements less than 23°. Such linearity allows us to calibrate the difference in stiffness (due to manufacturing error) between the 3 legs by linearly scaling the input voltage signal.

The primary calibration procedure involves following the lines trajectory (see Fig. 6) which includes linear motion of both inner and outer actuator motion for all three arms. From empirical data collected in Fig. 5, we observe the over-extension cause by our modeling underestimating the inner and outer actuation resistance in the uncalibrated path in Fig. 6. More importantly, each of the three lines have different lengths, meaning an actuator-specific multiplier proportional to their length error should be assigned to the voltage signal. To find the coefficient, we multiply each actuator's input voltage by $\sqrt{\frac{trajectory\ length}{actual\ length}}$, since displacement is linearly related to voltage squared (Fig. 5).

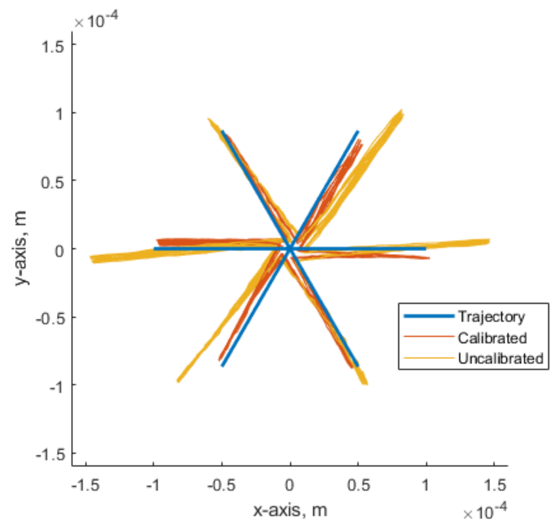


Fig. 6. Line trajectory tracing. Comparison between before and after actuator-specific calibration procedure. Improving the RMS accuracy by 3 times

C. Workspace and Trajectory

Desired trajectories were converted to end-effector positions and actuator input voltage signals were obtained as mentioned in section III-A. The position of the top platform is measured during operation and Fig. 7 shows the actual path traced by microDelta as compared to the kinematic model, where the RMS precision accuracy are listed in table II. The traced path closely resembles the programmed path, demonstrating the precise motion of compliant mechanisms

Fig. 8 illustrates the theoretical workspace of microDelta with a volume of 0.0016 mm^3. A slice of the workspace (red, upper figure) is converted to a trajectory (blue, lower figure)

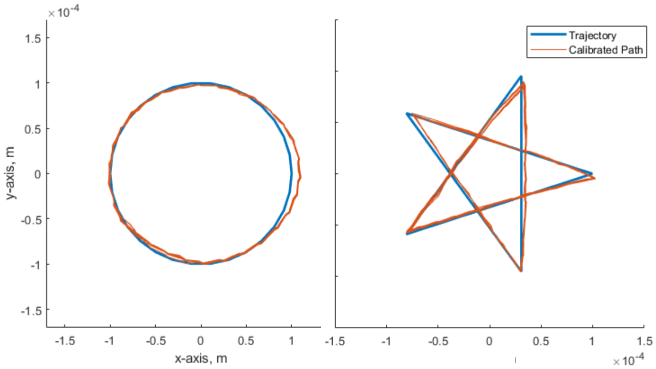


Fig. 7. Circle and star trajectory tracing. No further calibrations were attempted beyond the initial line pattern scaling process.

TABLE II

Trajectory following results. RMS precision and accuracy are calculated over five cycles of data recorded at 60 fps.

Trajectory	Frequency (Hz)	RMS Precision (µm)	RMS Accuracy (µm)
Circle	0.2	2.51	3.97
Star	0.2	4.65	2.98
Lines	0.2	11.02	5.34

and the calibrated following result is also plotted. Note that the theoretical workspace is the 2D surface boundary of actuator displacement sweep of $\pm 8°$, where the actuator is able to operate at voltages well below dielectric breakdown. Fabrication procedures that deposits oxide or insulating materials may increase the dielectric breakdown voltage and thus the safe workspace of microDelta.

VI. CONCLUSION

This work has demonstrated the ability for additive manufacturing at micro-scale to produce micro-robots with repeatability and precision. The unique size and workspace of this robotic system allows for micro-pick-and-place tasks with limited footprint, novel legged micro-robots, and haptics applications. Future directions of this work include closed-loop feedback control of the actuators to ensure accurate motion against creep, a time-dependent, visco-elastic strain.

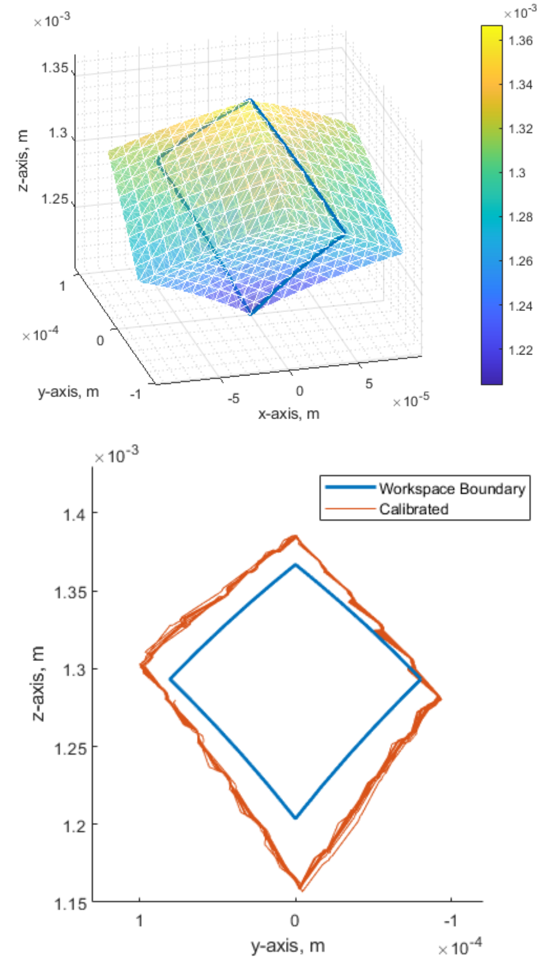


Fig. 8. Workspace of the microDelta with maximum input displacement of $\pm 8°$. The red outline is a slice of the workspace and the input trajectory of the lower image

REFERENCES

[1] R. Pierre, R. St, W. Gosrich, and S. Bergbreiter, "A 3d-printed 1 mg legged microrobot running at 15 body lengths per second," 05 2018, pp. 59–62.

[2] H. McClintock, F. Z. Temel, N. Doshi, J. sung Koh, and R. J. Wood, "The millidelta: A high-bandwidth, high-precision, millimeter-scale delta robot," *Science Robotics*, 2018.

[3] D. S. Contreras, D. S. Drew, and K. S. J. Pister, "First steps of a millimeter-scale walking silicon robot," in *2017 19th International Conference on Solid-State Sensors, Actuators and Microsystems (TRANSDUCERS)*, 2017, pp. 910–913.

[4] S. Kim and S. Bergbreiter, "Fabrication and characterization of 3d printed out-of-plane torsional comb-drive actuators for microrobotics," in *2021 21st International Conference on Solid-State Sensors, Actuators and Microsystems (Transducers)*, 2021, pp. 6–9.

[5] R. L. W. II, "The delta parallel robot: Kinematics solutions," Internet Publication, Ohio, January, 2016.

[6] L. Howell, S. Magleby, and B. Olsen, *Handbook of Compliant Mechanisms*. Wiley, 2013.

[7] S. Kim, C. Velez, R. S. Pierre, G. L. Smith, and S. Bergbreiter, "A two-step fabrication method for 3d printed microactuators: Characterization and actuated mechanisms," *Journal of Microelectromechanical Systems*, vol. 29, no. 4, pp. 544–552, 2020.

[8] S. Staicu and D.-C. Carp-Ciocardia, "Dynamic analysis of clavel's delta parallel robot," vol. 3, 09 2003.

[9] R. Norton, *Design of Machinery: An Introduction to the Synthesis and Analysis of Mechanisms and Machines*. McGraw-Hill Higher Education, 2003.

Guaranteed Parameter Uncertainty Estimation Using Interval Analysis for Robust Safety-Critical Control

Teodor Tchalakov[1], Qin Lin[2], and John M. Dolan[3]

***Abstract*— Robust control Lyapunov functions (CLFs) and control barrier functions (CBFs) use the largest bounds of parameter uncertainties to achieve robust safety-critical control, but the literature has proposed the bounds be manually tuned. This tuning problem requires balancing large uncertainty bounds in robust controllers that produce aggressive behavior against small uncertainty bounds that violate robustness. We present an over-approximate model parameter uncertainty estimation method for applications in robust CLF-CBF quadratic programming (QP) controllers using interval analysis. This work expands upon the set inversion via interval analysis (SIVIA) algorithm, which can be used offline to find bounds on parameter uncertainties from system observations. We improve the efficiency of SIVIA for finding the largest verified parameter intervals with an over-approximate heuristic search that uses system identification to provide initial guesses. The identification of model uncertainties can enable the life-long operation of robust safety-critical systems. We test our approach's accuracy and efficiency in comparison to SIVIA on an adaptive cruise control scenario with a robust CLF-CBF-QP controller.**

***Index Terms*— Parameter Uncertainty Estimation, Set Inversion, Autonomous Driving, Robust Control**

I. INTRODUCTION

Safety-critical systems require solving for verifiable controls that abide by safety constraints. A common method for provable safe control is control barrier functions (CBFs). CBFs require an accurate system model whose dynamics often need to have carefully identified parameters. If the parameters of the nominal/identified model are not representative of the true model when the controller is deployed, then the safety of the system can be violated [1]. In autonomous cars, the true model parameters change over time due to either environmental changes such as friction with the road or internal changes like the total mass of the car due to cargo being taken out or put in. Any effort to measure the real time model parameters will include uncertainty due to measurement noise. To handle model uncertainties, recent developments of robust control Lyapunov functions (CLFs) paired with CBFs in quadratic programming (QP) controllers have provided a framework to include uncertainties in a CLF-CBF-QP design that guarantees safety given a model's largest bounded uncertainty [2]. This work recommends that the upper uncertainty bounds be manually tuned, however, tuning requires choosing between a large uncertainty bound that will create an aggressive controller and a small uncertainty bound that might not contain all the true model uncertainties and therefore violate safety.

[1]Teodor Tchalakov is with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA, ttcha2@illinois.edu

[2]Qin Lin is with the Electrical Engineering and Computer Science Department, Cleveland State University, Cleveland, OH 44115, USA, q.lin80@csuohio.edu

[3]John M. Dolan is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA, jdolan@andrew.cmu.edu

In this paper, we investigate using model observations to solve for feasible parameter sets and then find a model's largest uncertainty bounds. Our approach expands upon the set inversion via interval analysis (SIVIA) [3] algorithm, which solves for feasible parameter sets offline. We propose solving for the minimally bounding intervals of a feasible parameter set in an over-approximate manner to improve computational efficiency for applications in robust CLF-CBF-QP controllers. Our search method uses heuristics and system parameter identification guesses to motivate the search towards regions of feasible parameters.

The rest of the paper is organized as follows. Section II reviews other approaches to model uncertainty in CLF-CBF-QP controllers and previous work in improving the efficiency of SIVIA. Section III provides background on CLF-CBF-QP, interval analysis, and SIVIA parameter estimation. Section IV proposes our over-approximate approach. Section V applies our approach to an adaptive cruise control scenario. Section VI outlines a future framework for parameter uncertainty estimation.

II. RELATED WORKS

Alongside the optimal robust safety-critical control method of [2] that uses predefined bounded uncertainties, other methods use statistical representations of model uncertainty and Bayesian neural networks such as Gaussian processes [4]–[6] to learn robust safety controllers. While [4]–[6]'s approaches enable the system to adapt its model online, the learning-based approaches assume that its model uncertainty distributions are Gaussian and learned model does not computationally scale well with state-dimension. These learning methods also require a large collection of quality data for training. Other learning methods such as that in [7] use reinforcement learning to learn model uncertainty online from observations and create a robust CLF-CBF-QP controller. It is possible for [7] to not converge on the true model uncertainties during online reinforcement learning due to early violations of safety constraints resulting in premature failure.

In SIVIA [3] the model parameter estimation problem is defined as solving a set inversion problem for a feasible parameter set from system observations with bounded

< TABLE OF CONTENTS

uncertainties. This method guarantees convergence on feasible parameter sets by a recursive search using bisection. However, [3]'s computing time increases exponentially with the number of parameters and is therefore not capable of running online to provide a robust CLF-CBF-QP controller from [2] a continuously updated uncertainty bound. Efforts in [8], [9] improve the computational efficiency of the set inversion problem using Taylor Models to solve for approximate parameter bounds and [8] uses domain reduction techniques to create simple strategies to avoid recomputing Taylor models. While these methods reduce the computational time of solving for feasible parameter sets, the search time is not fast enough for real time estimation. In comparison to SIVIA's interval based set representation, [8], [9] use ellipsoids and Taylor models to represent feasible parameter sets more generally while maintaining tightness. The problem of choosing a small subset of measurements from sequential system measurements is exploited in [10] to reduce the number of observations used when computing feasible parameter sets. [10] uses heuristics to order past measurements by their importance to solving for the feasible parameter set which results in a decrease in computation.

III. BACKGROUND

A. Input-Output Linearization of True and Nominal Models

Consider a control affine nonlinear system that represents the true model dynamics

$$\begin{aligned}\dot{x} &= f(x) + g(x)u, \\ y &= h(x),\end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the control input, and $y \in \mathbb{R}^m$ is the output of the system. To input-output linearize the system we differentiate the output y until the control input u appears explicitly. Most mechanical system dynamics have a relative-degree of 2, but we will outline the input-output linearization for any relative-degree r. Therefore the vector of the outputs for a relative-degree of r is

$$y^{(r)} = L_f^r h(x) + L_g L_f^{r-1} h(x) u, \tag{2}$$

where the functions $L_f^r h$ and $L_g L_f^{r-1} h$ are the r^{th}-order Lie derivatives [11]. Hence, $y^{(r)}$ is the vector of r^{th} derivatives of each output in y, and (2) indicates that no input in u appears at lower than the r^{th} derivative of each output. If $L_g L_f^{r-1} h$ is invertible, then

$$u(x,\mu) = u_{ff}(x) + (L_g L_f^{r-1} h(x))^{-1} \mu, \tag{3}$$

where $u_{ff}(x)$ is the feed-forward control input:

$$u_{ff}(x) = -(L_g L_f^{r-1} h(x))^{-1} L_f^r h(x), \tag{4}$$

and $\mu \in \mathbb{R}^m$ is the *auxiliary input*. This control law results in the input-output linearized system $y^{(r)} = \mu$, and a state transformation of $\Phi : x \to (\eta, z)$ can be defined, where:

$$\eta = [h(x)^\top, L_f h(x)^\top, ..., L_f^{r-1} h(x)^\top]^\top \tag{5}$$

$$z \in Z, Z = \{x \in \mathbb{R}^n | \eta(x) \equiv 0\} \tag{6}$$

This transformation enables the closed-loop dynamics of the system to be represented as a linear time-invariant system on the transverse coordinates η, and the zero-dynamics manifold Z. The true dynamics input-output linearized system is then:

$$\begin{aligned}\bar{f}(\eta) &= F\eta, \bar{g}(\eta) = G \\ \dot{\eta} &= F\eta + G\eta \\ \mu &= -K\eta\end{aligned} \tag{7}$$

with $F \in R^{mr \times mr}$ and $G \in \mathbb{R}^{mr \times m}$:

$$F = \begin{bmatrix} 0 & I_m & . & . & 0 \\ 0 & 0 & I_m & . & 0 \\ . & . & . & & \\ 0 & . & . & . & I_m \\ 0 & . & . & . & 0 \end{bmatrix}, G = \begin{bmatrix} 0 \\ . \\ . \\ 0 \\ I_m \end{bmatrix}, \tag{8}$$

When the input-output linearized system of the true model is unknown, the dynamics get designed using the nominal vector fields $\tilde{f}(x), \tilde{g}(x)$ and the precontrol law (3) gets reformulated as

$$\tilde{u}(x,\mu) = \tilde{u}_{ff}(x) + (L_{\tilde{g}} L_{\tilde{f}}^{r-1} h(x))^{-1} \mu, \tag{9}$$

where

$$\tilde{u}_{ff}(x) = -(L_{\tilde{g}} L_{\tilde{f}}^{r-1} h(x))^{-1} L_{\tilde{f}}^r h(x). \tag{10}$$

Then substituting in $\tilde{u}(x,\mu)$ from (9) into (2) yields

$$y^{(r)} = \mu + \Delta_1(x) + \Delta_2(x)\mu, \tag{11}$$

where

$$\begin{aligned}\Delta_1 &:= L_f^r h(x) - L_g L_f^{r-1} h(x) (L_{\tilde{g}} L_{\tilde{f}}^{r-1} h(x))^{-1} L_{\tilde{f}}^r h(x) \\ \Delta_2 &:= L_g L_f^{r-1} h(x) (L_{\tilde{f}}^{r-1} h(x))^{-1} - I_m.\end{aligned} \tag{12}$$

Likewise the dynamics of η from (7) now take the form:

$$\dot{\eta} = (F\eta + G\Delta_1(\eta, z)) + G(I_m + \Delta_2(\eta, z))\mu \tag{13}$$

If Δ_1 and Δ_2 are 0 then the uncertainty is zero and the nominal model correctly describes the true transverse dynamics of (7). Using the relationship between the true and nominal model allows us to bound the effects of uncertainty and to construct a robust CLF-CBF-QP controller as defined in [2], which uses the largest Δ_1 and Δ_2 values to provide guaranteed safety and stability for bounded uncertainty. It is suggested in [2] that the Δ_1 and Δ_2 values be manually defined based on the expected uncertainty of the system. Even though a safety controller does not have access to the true model which is required to solve for Δ_1 and Δ_2 exactly, we study the effects of how the resulting Δ values from guaranteed parameter uncertainty estimation can be applied to the robust CLF-CBF-QP controller.

B. Interval Analysis

1) Intervals and Boxes: An interval is a closed and bounded set of real numbers that describe all points within the interval $[x] \in \mathbb{R}$:

$$[x] = [x^-, x^+] = \{x | x^- \leq x \leq x^+\} \tag{14}$$

Multiple intervals can be used together in an interval vector or box to describe $\mathbb{R}^n$ as a cartesian product of its $\mathbb{R}$ intervals that is denoted as $\mathbb{IR}^n$:

$$[x] = [x_1^-, x_1^+] \times [x_2^-, x_2^+] \times ... \times [x_n^-, x_n^+] \tag{15}$$

2) Minimal Inclusion Function: A minimal inclusion function is defined for a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^y$ denoted as $[f]$, such that $[f] : \mathbb{IR}^n \rightarrow \mathbb{IR}^y$ where the result $[x] = [\{f(x)|x \in [x]\}]$ minimally bounds the output of the f over an input interval vector.

3) Minimally Bounded Model Uncertainty: The maximum values of Δ_1 and Δ_2 from equation (12) can be solved efficiently in a minimally bounded way using the minimal inclusion function on the nominal model's vector fields $\tilde{f}(x)$ and $\tilde{g}(x)$ and evaluating the minimal inclusion function with interval vectors that contain all of a model's feasible parameters.

C. Set Inversion via Interval Analysis

SIVIA [3] is a recursive bisection search algorithm that solves the following problem of set inversion:

$$\mathbb{P} = \{p \in \mathbb{R}^n | f(p) \in Y\} = f^{-1}(Y), \tag{16}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $\mathbb{P} \subset \mathbb{R}^n$, and $Y \subset \mathbb{R}^m$. This set inversion algorithm can be applied to parameter estimation as shown in Fig. 1 by assigning $\mathbb{P}$ as the set of unknown parameters, $[f]$ as the inclusion function of a model f, and Y as the observation space which consists of model measurements and their associated measurement uncertainty intervals. To validate if a parameter is a feasible parameter, the inclusion function of the model using the parameter guess must result in boxes that are consistent with all observations.

For instance, if a model f has an observation function $y = h(x, u, p)$ where $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the control input, $p \in \mathbb{R}^v$ are the unknown parameters, and $y \in \mathbb{R}^m$ is the measurement of the system, then an inclusion function can be defined for each system measurement. The state and control input measurements can be written as intervals $[x]$ and $[u]$ if there is uncertainty in the system input. Likewise an interval $[p]$ can be used as a parameter guess to evaluate a set of parameters together.

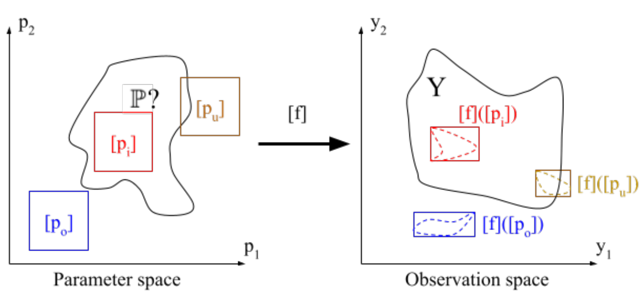


Fig. 1: Red boxes are feasible, Yellow are undetermined, and Blue are infeasible.

To converge upon the feasible parameters as outlined in Algorithm 1, SIVIA begins with an initial search box $[x](0) \subset \mathbb{P}$. This initial box is then evaluated on the observations of the system to check whether the resulting inclusion function $[f]([p])]$ produces a box for one of three cases inside of the observation space Y as outlined in Fig. 1: (1) $[x]$ is fully enclosed in Y and will be completely counted as a valid set of parameters, (2) is partially enclosed in Y and will need to continue to be searched by bisection, (3) the interval is does not intersect with Y and will therefore not be included in the set of parameters. Termination of the search in a box will happen if a feasible or infeasible box is found or an undetermined box reaches a width (the largest interval dimension) that is less than ϵ, at which point the desired accuracy has been met.

Algorithm 1: SIVIA

Input : $[x](0)$,f,Y,ϵ
Output: $L_{in}, L_{undetermined}, L_{out}$
1 $L = \{[x](0)\}$
2 pop $[x]$ from L
3 if $[f]([x]) \subset Y$, push $[x]$ into L_{in}
4 else if $[f]([x]) \cap Y = 0$, push $[x]$ into L_{out}
5 else if $width([x]) < \epsilon$, push $[x]$ into $L_{undetermined}$
6 else, bisect $[x]$ and push to L
7 if $L \neq 0$, go to 2

IV. Bounding SIVIA

A. Over-approximately Bounding Feasible Parameters

The exponential computational requirements of SIVIA limit it from being applied online. For this reason we propose Bounding SIVIA, which modifies SIVIA to focus on guaranteeing an over-approximation of feasible parameters that are minimally bounded by a box as outlined in Algorithm 2. While finding the smallest box that encloses a feasible parameter set can be achieved after running SIVIA without modification, we exploit the problem structure to skip regions during the search that are within a growing box and therefore seek to benefit in decreased computational time. We also propose motivating the search towards regions that contain the bounding feasible parameters by applying heuristics to our algorithm's search order. Our method will contain all feasible parameters as it only changes the order in which SIVIA evaluates boxes and skipped regions do not effect the convergence of the algorithm as the recursive evaluation of an individual box that is the result of bisection does not depend on any other boxes.

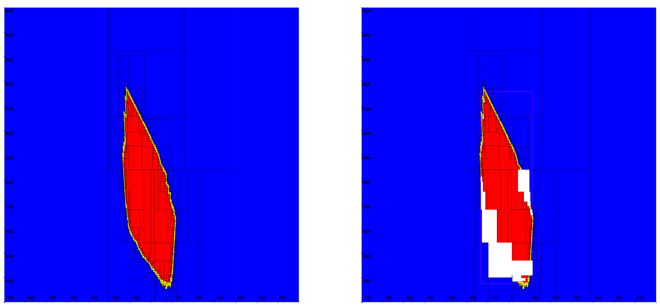
B. System Parameter Identification Initial Guess

The SIVIA algorithm begins with a large predefined search space which it can quickly reduce the volume of as it bisects, but SIVIA has difficulty evaluating the boundaries between feasible and infeasible quickly [3]. To motivate the search to find regions that will grow the enclosing box, a heuristic that uses an initial guess is used as a starting point to search for a possible location that might contain feasible parameter sets. If being applied offline, any general

Algorithm 2: Bounding SIVIA

Input : $[x](0)$,f,Y,ϵ,heuristic
Output: $L_{in}, L_{undetermined}, L_{out}$, LargestBox

1 $L = \{[x](0)\}$
2 LargestBox = [empty]
3 pop $[x]$ from L
4 if $[f]([x]) \subset Y$, push $[x]$ into L_{in}, LargestBox = LargestBox $\cup$ $[x]$
5 else if $[f]([x]) \cap Y = 0$, push $[x]$ into L_{out}
6 else if $width([x]) < \epsilon$, push $[x]$ into $L_{undetermined}$
7 else, bisect $[x]$ and exclude $[x]$ if in LargestBox otherwise push $[x]$ to L
8 sort L based on heuristic
9 if $L \neq 0$, go to 2



(a) SIVIA (b) Bounding SIVIA

Fig. 2: Bounding SIVIA skips regions as the search grows the largest box of the feasible parameters, while SIVIA checks all regions.

system model-fitting technique such as least-squares fitting or gradient-based optimization [12] will produce good guesses if there is a variety of data. However, online model regression for system identification requires solving problems of data assimilation where there might not be enough data to fit with respect to a state variable or the clustering of samples can produce regions of high weight in the regression process while ignoring others. For this paper, we assume that an accurate system parameter identification has been made near the true model parameters. We recommend exploring for online use, any dual estimation Kalman filter such as [13].

C. Heuristics-Based Search

The goal is to find the smallest box enclosing all parameters that are consistent with measurements of the system. The Bounding SIVIA algorithm does not need to check boxes that are already inside the current union of already valid parameters, as this would not increase the size of the smallest box enclosing all parameters. Therefore we can remove those boxes as we are bisecting. This means that it would be best to find boxes that are far away from each other such that the largest feasible box grows. Once a heuristic is decided, the order in which the available search spaces are explored in Algorithm 1 can be sorted based on the heuristic as they are added. A heuristic can be combined with a system identification guess to make a nearest and farthest box heuristic based off the distance between the box and the guess. This distance can be calculated as a geometric distance from the guess and the box centroid, where the centroid is computed as the vector containing the averages of each interval dimension along a box. We also investigate using a simple volume heuristic that is the product of the ranges of a box's intervals.

V. Adaptive Cruise Control Scenario Numerical Evaluation

A common problem concerning robust CLF-CBF-QP controllers is in car safety. We will use the following modelled scenario of adaptive cruise control (see Fig. 3) to demonstrate interval analysis on finding the largest Δs and apply our bounded SIVIA approach. We define the dynamics as follows:

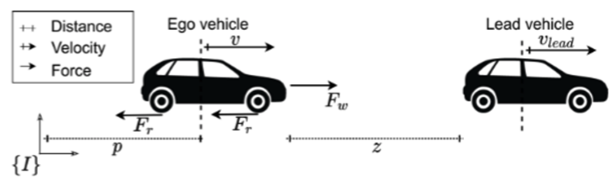


Fig. 3: Adaptive Cruise Control Scenario

$$x = [p, v, z]^\top \in R^3 \tag{17}$$

$$\dot{x} = \begin{bmatrix} v \\ -F_r(v)/m \\ v_{lead} - v \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \\ 0 \end{bmatrix} u = f(x) + g(x)u \tag{18}$$

$$F_r(v) = f_0 + f_1 v + f_2 v^2 \tag{19}$$

$$h(x) = z - T_h(v) - .5(v - v_{lead}^2)/(c_d g) \tag{20}$$

where m, f_0, f_1, and f_2 are unknown parameters and represent the car's mass and the wheel resistance respectfully. $h(x)$ defines the safety requirement of the car as maintain a headway T seconds behind the leading vehicle based on the velocity of the car and its ability to slow down to the lead vehicle's speed.

Our numerical experiment to test the computational efficiency of our approach is done with a simplified version of the adaptive cruise control scenario that has only the m and f_0 parameters as unknown while the rest are known. We evaluate SIVIA and Bounding SIVIA with all heuristics on 10 simulated data sets of 300 uniformly sampled measurements where the error of the output acceleration and input velocity of the system from equation (18) have a standard deviation of $1/30$ per each units. SIVIA and Bounding SIVIA then had their accuracy value ϵ set to 10 and the assumed measurement error be bounded by .1 for acceleration and velocity. We present performance metrics for the average convergence time and the number of boxes evaluated till convergence.

< TABLE OF CONTENTS

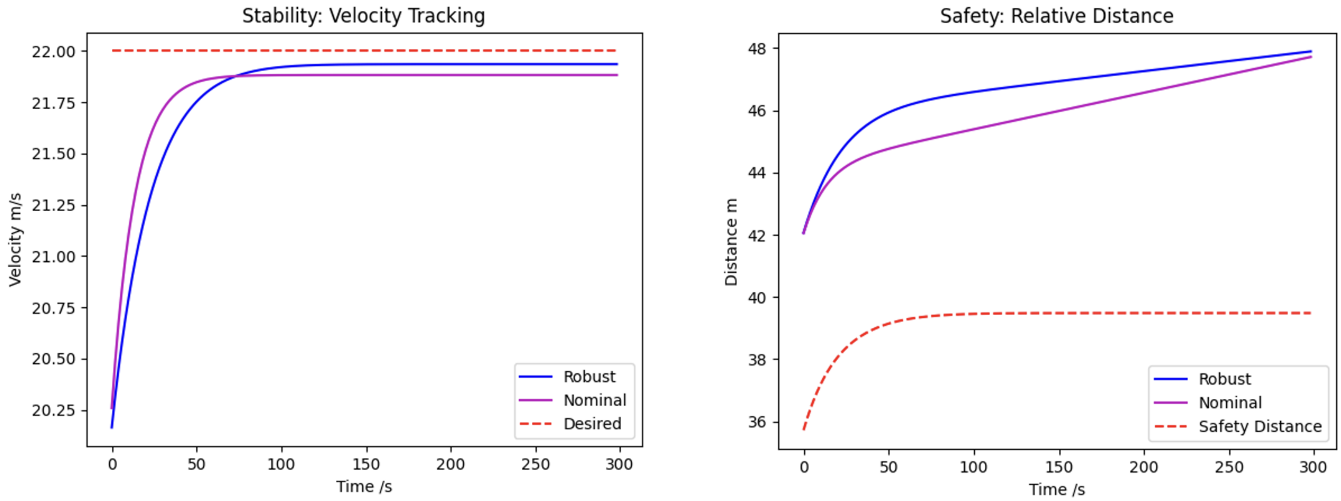


Fig. 4: Nominal vs Robust Bounding SIVIA Adaptive Cruise Control Scenario. Nominal model parameters: $m = 1155$, $f_0 = .05$, $f_1 = 5$, $f_2 = .25$. True model parameters: $m = 1650$, $f_0 = .1$, $f_1 = 5$, $f_2 = .25$.

TABLE I: Bounding SIVIA and Heuristic Performance

Bounding SIVIA		
Heuristic (n=10)	**Average Time** (s)	**Average Evaluations** (Boxes)
None	7.03	216
Volume	5.31	120
Nearest	7.44	208
Farthest	5.92	147
SIVIA [3]		
Baseline	9.79	340

A. Results

Our experiment was run on a 2021 M1 Macbook Pro and implemented in Python using the PyIbex library [14]. The results in Table I show that adding a heuristic will help improve the bounded search efficiency when compared to using no heuristic and that skipping search regions decreases the number of boxes evaluated on average from 340 boxes to a worst case of 216 boxes and improves the average computation time by 28.1%. Although searching from the center outwards from the system identification guess did not perform as well as searching from outwards in. Because these values are still on the order of seconds, our algorithm did not speed up the computation enough for application in real time safety-critical motion planning.

We can then apply Bounding SIVIA's minimally enclosing feasible parameter box by solving for the largest Δ bounds as outlined in Sec. III-B3 for the adaptive cruise control scenario. For the particle example in Fig. 4, the bounds for m and f_0 found by Bounding SIVIA are $[m] = [1544.75, 1773.85]$ and $[f_0] = [91.8, 74.224]$ while the true model parameters were $m = 1650$ and $f_0 = .1$. Fig. 4 demonstrates that compared to a nominal model without robust uncertainty assumptions, the velocity tracking stability of the robust safety controller is closer to the target velocity. Likewise both controllers maintain the relative safety distance. It is therefore possible that Bounding SIVIA can tune a CLF-CBF-QP controller to behave without aggressive changes in velocity or relative safety distance as expected for overly large Δs.

VI. Discussion

A. Explicitly Learned Feasible Parameters

We have discovered many areas that the problem of guaranteed parameter uncertainty estimation can be modified to improve run time performance. In this section we would like to outline a new approach that utilizes more exact and efficient set representations, skips the necessity to search a parameter space by intersecting undetermined parameter sets to find feasible parameter sets. This new approach is done by explicitly computing a dataset of undetermined feasible parameter sets at any desired granularity using SIVIA offline, fitting a constrained polynomial zonotope [15] tightly over each undetermined feasible parameter set, and then training a neural network to learn the mapping of a system's input and outputs with associated uncertainty bounds to a constrained polynomial zonotope. Then online, the measurements of the system can be passed to the neural network to compute the approximate constrained polynomial zonotope and all of the undetermined feasible parameter sets can then be intersected to find the true feasible parameter set. The main challenges of this approach are creating a training dataset for a given model and uncertainties when the model is high dimensional, tuning the neural network to balance between accuracy and evaluation speed, creating a method to fit a constrained polynomial zonotope over a list of intervals from SIVIA's results, and finding an efficient manner to sample or compute a model's Δ values from the feasible parameter set represented by the intersected constrained polynomial zonotope.

VII. Conclusion

Our approach can identify the smallest box that includes all parameters but it is not able to do it online. The application of this guaranteed approach to higher dimensional unknown parameter systems will require a more effective algorithmic search that can scale efficiently. In future works we would like to explore the use of more tightly bounding polytope representations and computing the problem offline to then solve for the intersection of feasible parameters online. We would also like to experimentally evaluate this approach on a physical system and investigate its performance on rapidly changing model parameters.

VIII. Acknowledgment

Special thanks to Dr. Qin Lin, Dr. John M. Dolan, and Rachel Burcin for working to uphold research opportunities in robotics through the RISS 2022 experience and bringing everyone to an in person experience. The authors gratefully acknowledge the support of the NSF CMU Robotics Institute REU Site, Award #1950811 in performing this work.

References

[1] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

[2] Q. Nguyen and K. Sreenath, "Robust safety-critical control for dynamic robotics," *IEEE Transactions on Automatic Control*, vol. 67, no. 3, pp. 1073–1088, 2022.

[3] L. Jaulin and E. Walter, "Set inversion via interval analysis for nonlinear bounded-error estimation," *Automatica*, vol. 29, no. 4, pp. 1053–1064, 1993.

[4] D. D. Fan, J. Nguyen, R. Thakker, N. Alatur, A.-a. Agha-mohammadi, and E. A. Theodorou, "Bayesian learning-based adaptive control for safety critical systems," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4093–4099.

[5] S. Bansal, R. Calandra, T. Xiao, S. Levine, and C. J. Tomlin, "Goal-driven dynamics learning via bayesian optimization," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 5168–5173.

[6] R. Cheng, M. J. Khojasteh, A. D. Ames, and J. W. Burdick, "Safe multi-agent interaction through robust control barrier functions with learned uncertainties," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 777–783.

[7] J. Choi, F. Castañeda, C. Tomlin, and K. Sreenath, "Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions," 07 2020.

[8] R. Paulen, M. E. Villanueva, and B. Chachuat, "Guaranteed parameter estimation of non-linear dynamic systems using high-order bounding techniques with domain and cpu-time reduction strategies," *IMA Journal of Mathematical Control and Information*, vol. 33, no. 3, pp. 563–587, 2016.

[9] B. Chachuat, B. Houska, R. Paulen, N. Peri'c, J. Rajyaguru, and M. E. Villanueva, "Set-theoretic approaches in analysis, estimation and control of nonlinear systems," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 981–995, 2015, 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015.

[10] P. Artzová and R. Paulen, "Moving-horizon guaranteed parameter estimation," *IFAC-PapersOnLine*, vol. 52, no. 1, pp. 112–117, 2019, 12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS 2019.

[11] S. Sastry, "Nonlinear systems: analysis, stability, and control," vol. 10, 1999.

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[13] E. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 2000, pp. 153–158.

[14] D. Benoit and J. Luc, "Pyibex." [Online]. Available: http://codac-team.github.io/pyIbex/sphinx/index.html

[15] N. Kochdumper and M. Althoff, "Constrained polynomial zonotopes," 2020.

Detecting and classifying bus stop trash cans using camera-equipped public transit vehicles

Tim Storm[1] and Christoph Mertz[2]

***Abstract*— Trash cans are a central tool in managing the disposal of trash in urban areas, but require human supervision to ensure regular emptying. It is difficult to manage a large number of waste bins spread across a whole city, which presents an opportunity for computer vision technology to identify cans that require attention without human intervention.**
Previous work has leveraged a camera-equipped bus to deploy a single deep learning based computer vision model to detect trash cans along the path of the bus and classify their fill level. We improve upon their work by presenting a multi-stage pipeline that combines their detection model with a separate, second model trained purely for classification. The detector will identify and cut out trash cans from an image, which will then be classified as either "Empty", "Full" or "having a garbage bag next to it". Our approach significantly increases the overall accuracy and precision for both tasks, as calculated by the commonly used COCO metrics. Additionally, we present a lightweight variant of our detection model, which can be run on the bus itself, where only limited computational resources available. This enables us to actually deploy our system in a near real-time setting.

***Index Terms*— Computer Vision for Transportation, Intelligent Transportation Systems, Environment Monitoring and Management, Object Detection, Segmentation and Categorization**

I. INTRODUCTION

The comprehensive report "What a Waste 2.0" published in 2018 estimates that by 2050 waste generation rates will outpace population growth by a factor of two, which will pose large challenges in the solid waste management sector, especially in low-income countries [1].
This is a considerable point of costs in many cities: According to that same report, solid waste management takes up to 19 percent of a city's budget, depending on the area. On average 60 to 70 percent of total operational costs can be accounted to the task of waste collection.
Even in high-income areas waste collection typically relies on designated garbage pickup schedules and human supervision, but these methods are highly inflexible in adjusting to short-term trends and scaling to large areas. These can be serious problems when demand for garbage collection grows, as cities and population increase in size.
Planning-based systems can help in determining effective routes to help garbage disposal companies attend to the trash cans that actually need to be emptied, but they require information about the state of trash cans in an area.

[1]Tim Storm is with University of Paderborn, Paderborn, Germany `tistorm@mail.upb.de`
[2]Christoph Mertz is with the NavLab, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA `cmertz@andrew.cmu.edu`

To address this problem, we implement a system [1], which uses camera-equipped public transport buses to continuously monitor trash cans along a bus route. This approach utilizes existing infrastructure and ensures reasonably accurate measurements through regular traversal of the same areas.
We aim to deploy this system on the BusEdge platform [2]. BusEdge controls a number of sensors and safety cameras on the bus itself and allows accessing and working with the resulting data without any additional hardware, by offloading the computational load to a nearby Cloudlet instead.
In particular we are presenting a three-stage detection-and-classification pipeline, which is able to detect trash cans along the side of the road and estimate their fill level based on their outward appearance, while respecting the resource restrictions imposed by BusEdge.
To do so, we apply multiple computer vision models based on the ResNet architecture [3] trained with Detectron2, an open source library that provides a number of state-of-the-art object detection algorithms [4].
This is a continuation of previous work [5], which proposed this pipeline and implemented parts of it. We provide an analysis of commonly accepted metrics like precision and recall, and show how our changes improve the previous results.
Once deployed our pipeline can assist an employee of the bus company by automatically providing an overview of all existing trash cans, monitoring their fill level over time and suggesting bus stops where trash cans are or will soon be too full.

II. RELATED WORK

A. IoT-based Trash level monitoring

In order to improve effective emptying of waste containers a number of Internet of Things (IoT) solutions have been proposed [6]–[9], that aim to build "smart" trash cans, equipped with sensors and a network connection, which allow for reliable remote monitoring. These systems can track trash levels much more accurately than a computer vision approach, but they have the distinct disadvantage of requiring specialized hardware.
This makes retrofitting a city with such devices a tedious endeavor and requires additional maintenance, which makes scaling up difficult.
Our approach uses existing public infrastructure, which makes deployment easier and more flexible.

[1]All code is available on https://github.com/timlst/riss-trash-detection

< TABLE OF CONTENTS

B. Computer vision for trash detection

In the past there have been many attempts to utilize computer vision methods to fight increased pollution, both in terms of localizing and classifying it. [10], [11]

Machine learning techniques such as Support-vector machines (SVMs) [12] or Convolutional Neural Networks (CNNs) [13], [14] were used to detect littering in both urban, as well as aquatic enviroments [15]. The TACO [16] data set offers a good starting point for detecting waste left in nature. Other research directions include classifying trash to help with recycling [17], [18].

While such methods certainly help with improving municipal cleanliness, they do not account for waste, that has been properly disposed of and that is ready for collection.

C. Deep Learning in computer vision

In recent times convolutional neural networks have been out-performing more classical computer vision algorithms, which has shifted the field towards heavy use of deep learning based methods.

A number of frameworks [4], [19], [20] at varying degrees of abstraction have opened up the field by providing pre-trained models for different architectures and applications.

The task of object detection in particular allows for a wide range of combinations: We will be using the Faster R-CNN architecture of the R-CNN family [21]–[23], a two-stage detector with an emphasis on accuracy, readily available within our framework of choice - Detectron2 [4]. Depending on the application, many different backbones can be used for feature extraction. A common choice are entries from the ResNet family [3], but other choices such as Inception [24], [25] or combinations of the two [26] exist.

These architectures, while highly accurate, tend to be quite slow, which is unsuitable for mobile and embedded devices. This has created other approaches with an emphasis on speed and efficiency. These range from more efficient backbones like the MobileNets family [27] to a class of single-stage detectors like SSD [28], YOLO [29] or RetinaNet [30], that aim for (near) real-time application.

III. OVERVIEW

We propose a three-stage pipeline in order to detect trash cans in a real-time context: On-the-edge preselection, Trash can detection and Trash can classification. The pipeline is designed to be deployed on the BusEdge platform proposed by [2]. This section largely recaps the previous work by [5].

A. BusEdge

BusEdge [2] provides a framework for us to work with data collected by sensors on a common public transportation bus. This includes information such as images from exterior cameras, GPS, acceleration etc. Notably the bus itself is equipped with a computer not capable of (and not designed to) executing computationally intensive tasks. Instead a typical BusEdge pipeline uses only lightweight filters that run on the bus itself. Data that passes through the filter is then sent to a more capable "cognitive engine" via wireless network, where a thorough analysis is possible.

B. On-the-edge preselection

The large amount of incoming data from the cameras makes it infeasible to consider every frame equally. Instead we want to concentrate our efforts on a promising subset of all incoming data.

The largest limitation in this context are the limited computational resources available: The limited bandwidth requires us to perform this preselection on the bus itself, since we cannot pass all images onto the cloudlet server.

We can address this issue by applying a lightweight detection model to roughly detect possible trash cans and then analyze the candidates more precisely later. Instead of being very precise, we are aiming for high recall at this stage. This leaves us with images that may contain a trash can, while discarding the ones that clearly do not. The whole step has to be performed reasonably fast, even with the lack of a dedicated graphics processing unit (GPU) typically available in a machine learning setting, so that the whole pipeline can run in near real-time.

For that reason, we want to look at efficient models specially designed for the deployment on mobile devices such as MobileNets [27] to account for the bus' hardware limitations.

C. Detection

Our main goal in this stage is to accurately detect and localize trash cans, regardless of their trash level, in the previously identified images.

This step will be performed on the cloudlet server, which is why we can apply more complex and computationally expensive models like RetinaNet [30] or ResNet [3] to achieve highest possible accuracy.

Previous experiments [5] have shown that, while architectures like Faster R-CNN are able to both localize and classify, the overall accuracy decreases heavily when relying on a combined model.

That is why we cut out the identified bounding boxes and feed them into a separate final stage instead.

D. Classification

The final stage of our pipeline is responsible for classifying the cropped images into three separate categories:

Trash cans that do not visibly contain trash will be considered empty for our purposes, as they are not in need of immediate attention. Trash cans that do visibly contain trash are considered (soon-to-be) full for our purposes. Since the bus camera can only see the outside of the cans, visible trash is an indicator of reaching maximum capacity. That means they will be ready to be emptied in the near future and are therefore of interest to the bus company. Our third class covers trash cans with a garbage bag in their immediate vicinity, regardless of their trash level, as garbage bags are indicative of a can that needs immediate attention, since trash is piling up beyond the contents of the can. To account for this class the cropped out bounding boxes are extended into all directions to include the surrounding area.

Since we can rely on human assistance for ambiguous images, we will assign a fourth class for predictions that fall

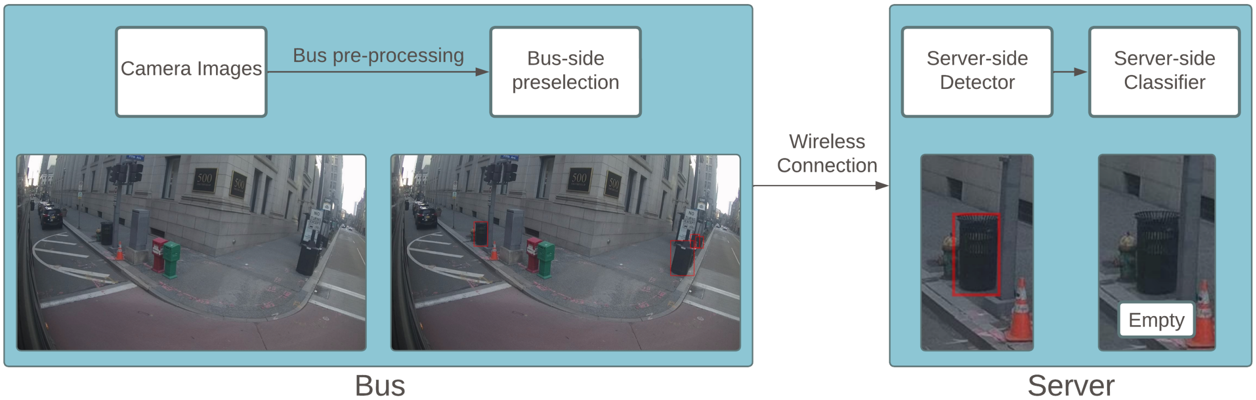


Fig. 1: Proposed detection and classification pipeline

below a certain confidence threshold. Under the (reasonable) assumption that a human operator will always be accurate, this option presents a trade-off between human effort and overall performance by relying on the operator's judgement for all predictions under the threshold.

E. Notifcation

In order for the proposed pipeline to be useful to the bus company, an additional stage is required: The responsible party has to be notified about the location of trash cans, which we deem to need attention. The BusEdge platform offers GPS data, which will be used for this purpose.
Our system provides all the necessary information for such an application to be built on top of it.

IV. METHODOLOGY

A. Data sets

1) Regarding existing data sets: We adapted existing data sets [31], [32] from previous work [5] for our purposes:
The data sets consist of 14,981 images of which 2682 images contain 3909 annotations. In the first case, the annotations assign a single label to every object of interest, whereas in the second case, they distinguish between "Full", "Empty" and "Garbage Bag".
It has to be noted, that the annotations used here do not distinguish between between domestic (movable) trash cans and permanently installed trash cans. We used the full provided data set for training purposes, but since we are mainly interested in permanent installations at bus stops, we disregard any failure cases related to other trash cans in our analysis (unless explicitly stated otherwise).

2) Detection data set: To train and test our detection models, we used the unaltered data set, as available at [31].

3) Classification data set: To train and evaluate our classification model, we had to make a few changes to the provided data set [32]. Since this task does not expect bounding boxes, we cut out the annotated bounding boxes from the existing images to use instead. To account for the "Garbage Bag" label, we extended the given bounding box by 25% horizontally and 10% vertically in each direction. If this extended bounding box contained a garbage bag annotation we assigned the appropriate "garbage bag" label, falling back to the original annotation otherwise.
This method produced a number of images not even classifiable by humans, which is why we then filtered out all images smaller than 32 pixels in either dimension (considered "small" by COCOeval [33]) For the reasons outlined in paragraph IV-A.1, we mainly look at a variant of our test set, which only contains bus stop trash cans.

B. Model training

We used our data sets as described in the previous section to finetune models from the ResNet family [3], as well as the MobileNet family [27]. All used models have been pre-trained on the ImageNet [34] data set.

1) Detection: Our server-side detector was trained using the Resnet101+FPN backbone for Faster R-CNN as proposed in [35], available within the Detectron2 model zoo [4], which we trained for approximately 30,000 iterations, while evaluating the validation set every 1,000 iterations.
The bus-side detector relies on the same architecture, but we compare different, simpler backbones capable of running on mobile devices, including Resnet18+FPN and MobilenetV2.

2) Classification: Our classifier is based on Resnet101 again: We only replaced the final fully-connected layer with a smaller fully-connected layer to account for our three classes. We also applied the Softmax function, so that the output becomes a probability vector.
We trained this model for around 10 epochs.

C. Evaluation

1) Detector: We report COCO-style metrics [33], as well as a precision-recall curve to evaluate the performance of our detection models.
Our use case needs only a very rough localization within an image, which is why we give special attention to the metrics

at an Intersection over Union (IoU) value of 50%. For the bus case we additionally benchmark the inference time in a CPU-only setting averaged across multiple runs.

2) Classifier: To evaluate classification performance, we provide overall precision and recall values for our three desired classifications on the bus stop-only test data. These are be partially visualized in a confusion matrix to highlight possible failure cases.
Additionally we compared these metrics given different confidence thresholds for human intervention as described in section III-D.
Calculating the CLASS-BALANCED-ACCURACY [36] allows us to a give a single, comparable value here, while accounting for the class imbalance of the data set.

V. RESULTS

A. Server-side Detection

When applied to our test set, we achieved an overall average precision of 78.3% at an overall average recall of 85.0%. The precision increased to 89.2% when looking at IoU 0.50.
We also saw a significant increase of over twenty percentage points across all categories, compared to the Retinanet model presented by [5] (see Table I).

The precision-recall curve shows promising results across all threshholds (see blue curve in Figure 2). This becomes even more apparent, if we limit ourselves to large (greater than 96 by 96 pixels) bounding boxes (see orange curve in Figure 2). These are especially interesting to us, since we expect to get close-up images of all bus stop trash cans when driving past them.

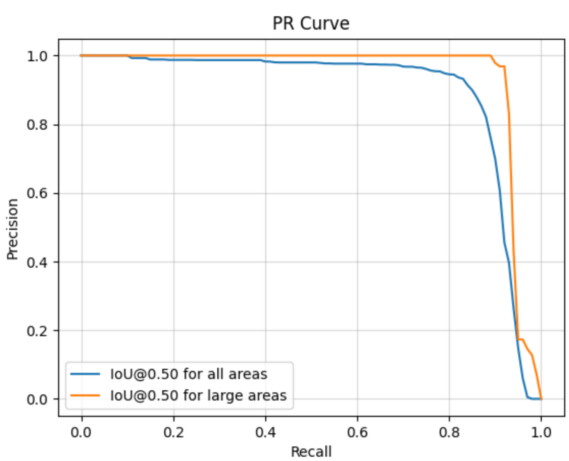


Fig. 2: Better model performance in the especially valuable "large" case

However there is a small amount of trash cans that we never manage to detect. By looking at these images, we managed to identify a failure cases: Occlusion.
We expect each trash can to appear in full view at least once, but as Figure 3 shows this may not always the case.

Fig. 3: Occluded trash can (green bounding box) is only visible for a single frame. In the next recorded frame (bottom), the bus has already passed the trash can.

B. Bus-side detection

Finetuning both backbones showed that the Resnet18 backbone surpassed the MobileNets-V2 variant in all our core categories (see table I): While there was a slight difference in inference time between the two, the difference is not large enough to make up for the additional accuracy.

A look at the precision recall curve (Figure 4) shows excellent performance of our lightweight model, when considering "large" trash cans, rivaling the recall of the server model with only slightly lower precision.

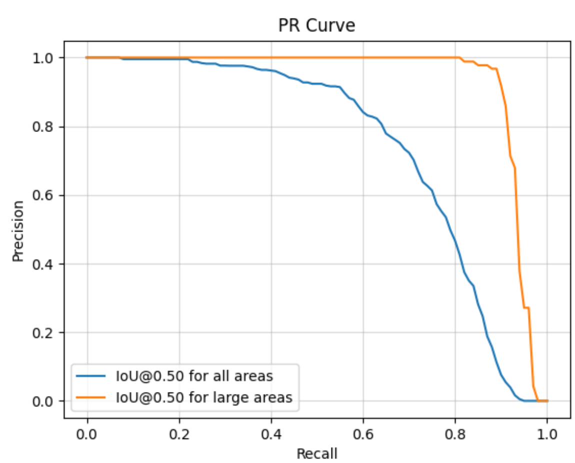


Fig. 4: Lightweight model performs exceptionally well in "large" case, even compared with the server-side detector

TABLE I: Resnet18 achieves higher results across the board

Backbone	AP	AP50	APm	APl	Inference (on CPU)	Inference (on GPU)
Resnet101-FPN	78.3%	89.2%	84.16%	88.55%	3111ms	152ms
RetinaNet	44.2%	67.4%	57.3%	79.2%	N/A	N/A
Resnet 18-FPN	50.13%	**74.41%**	61.28%	77.09%	604ms	34ms
MobileNetsV2	35.09%	55.36%	49.68%	62.58 %	556ms	37ms

By applying the model as a pre-selector at a threshold of 0.5, our combined data set of 14981 individual frames was cut down to a set of 2191 candidate frames (approximately 14.6%), while retaining close to 95% of all large instances.

C. *Classification*

Evaluating the classifier on our reduced test set (only including bus stops), we see great results in the Garbage Bag and Empty category, with the weakest performance in the Full category (see table II).

This equates to an BALANCED-ACCURACY of 0.904.

TABLE II: Performance of our classifier on the reduced test set

Label	Precision	Recall	Support
Empty	98.52%	98.52%	270
Full	86.96%	90.91%	22
Garbage Bag	98.90%	97.83%	92

The confusion matrix (Fig. 5) highlights this: Full trash cans are mistakenly identified as empty around ten percent of the time. While these are acceptable results, the bus company is especially interested in these trash can.

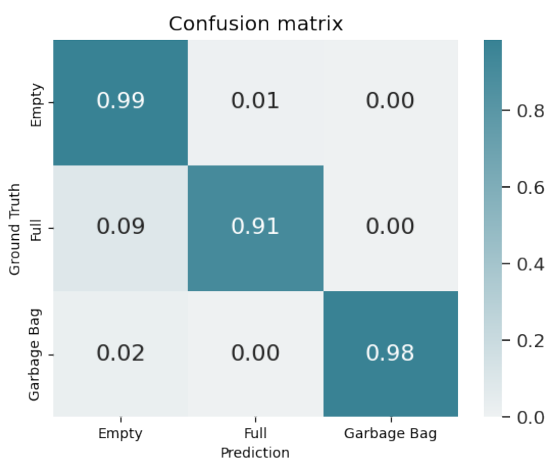


Fig. 5: One tenth of full trash cans are falsely predicted to be empty.

We found that the results can be improved by applying a minimum confidence threshold (as described in section III-D). We found the best results at a threshold of 87% (see Figure 6): With human intervention, we are able to correctly identify all full trash cans, while barely affecting the results

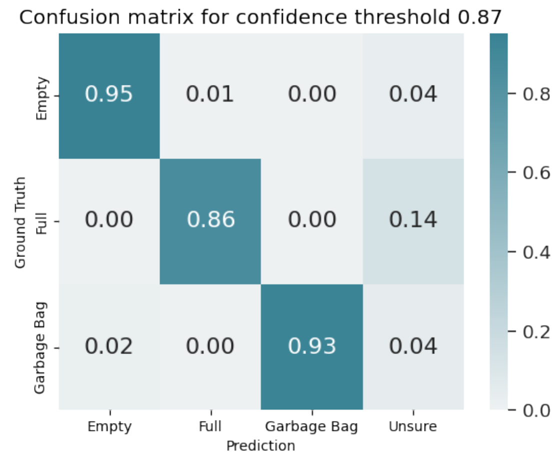


Fig. 6: A threshold of 87% allows us to find all full trash cans.

in the other categories. This translates to an improved BALANCED-ACCURACY score of 0.953. As a trade-off we required the operator's judgement for 15 images, which accounts for around 4 percent of all 384 images in the data set.

We found a similar trend when applying the classifier to the full test set (including every kind of trash can), but saw a significant increase in the fraction of misidentified full cans, where the confidence threshold proved to be less effective (see Appendix Figure 7). While there is a definite area of improvement, it serves as evidence, that our approach can be adapted for other kinds of waste containers.

VI. CONCLUSIONS AND FUTURE WORK

We presented a three stage pipeline for detecting and classifying trash cans, designed for deployment in an edge-computing context. Our model significantly outperforms previous results presented on this problem and the results indicate satisfying results in a real-world application. We also provide evidence, that our results may generalize to a wider variety of waste containers.

So far we have not tested our system in actual deployment. While we have tried to account for the limited resources, further experimentation is still required. Depending on the results we may want to consider other architectures such as SSD [28] or YOLO [29] in the pre-selection step.

ACKNOWLEDGMENT

Tim thanks Anurag Ghosh for his many helpful suggestions. Tim thanks Christoph for his mentorship and general guidance. Tim thanks Rachel and John for being able to participate in the RISS program, as well as the DAAD for its generous financial support, which made the experience possible in the first place. Data and background software were provided by projects sponsored in part by NSF under Award No 2038612 and Carnegie Mellon University's Mobility21 National University Transportation Center, which is sponsored by the US Department of Transportation.

REFERENCES

[1] S. Kaza, L. C. Yao, P. Bhada-Tata, and F. Van Woerden, *What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050*. Washington, DC: World Bank, Sept. 2018, accepted: 2018-08-31T16:25:22Z. [Online]. Available: https://openknowledge.worldbank.org/handle/10986/30317

[2] C. Ye, "BusEdge: Efficient Live Video Analytics for Transit Buses via Edge Computing," *The Robotics Institute Carnegie Mellon University*, July 2021. [Online]. Available: https://www.ri.cmu.edu/publications/busedge-efficient-live-video-analytics-for-transit-buses-via-edge-computing/

[3] "Deep Residual Learning for Image Recognition," Dec. 2015, arXiv:1512.03385 [cs]. [Online]. Available: http://arxiv.org/abs/1512.03385

[4] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," 2019. [Online]. Available: https://github.com/facebookresearch/detectron2

[5] E. Rotondo and C. Mertz, "Detecting and Classifying Waste Bin Garbage Levels Along Transit Bus Routes," *RISS Working Journal*, 2021.

[6] Y. Zhu, G. Jia, G. Han, Z. Zhou, and M. Guizani, "An NB-IoT-based smart trash can system for improved health in smart cities," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, June 2019, pp. 763–768, iSSN: 2376-6506.

[7] S. R. J. Ramson, S. Vishnu, A. A. Kirubaraj, T. Anagnostopoulos, and A. M. Abu-Mahfouz, "A LoRaWAN IoT-Enabled Trash Bin Level Monitoring System," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 786–795, Feb. 2022, conference Name: IEEE Transactions on Industrial Informatics.

[8] S. Kristanto, T. Yashiro, N. Koshizuka, and K. Sakamura, "Dynamic Polling Algorithm for Low Energy Garbage Level Measurement in Smart Trash Bin," in *Proceedings of the Second International Conference on IoT in Urban Space*, ser. Urb-IoT '16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 92–94. [Online]. Available: https://doi.org/10.1145/2962735.2962748

[9] H. Sohail, S. Ullah, A. Khan, O. B. Samin, and M. Omar, "Intelligent Trash Bin (ITB) with Trash Collection Efficiency Optimization using IoT Sensing," in *2019 8th International Conference on Information and Communication Technologies (ICICT)*, Nov. 2019, pp. 48–53.

[10] R. J. Firby, R. E. Kahn, P. N. Prokopowicz, and M. J. Swain, "Collecting Trash: A Test of Purposive Vision," p. 20.

[11] Y. Wang and X. Zhang, "Autonomous garbage detection for intelligent urban management," *MATEC Web of Conferences*, vol. 232, p. 01056, 2018, publisher: EDP Sciences.

[12] I. Salimi, B. S. Bayu Dewantara, and I. K. Wibowo, "Visual-based trash detection and classification system for smart trash bin robot," in *2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, Oct. 2018, pp. 378–383.

[13] M. S. Rad, A. von Kaenel, A. Droux, F. Tieche, N. Ouerhani, H. K. Ekenel, and J.-P. Thiran, "A Computer Vision System to Localize and Classify Wastes on the Streets," in *Computer Vision Systems*, ser. Lecture Notes in Computer Science, M. Liu, H. Chen, and M. Vincze, Eds. Cham: Springer International Publishing, 2017, pp. 195–204.

[14] N. Jayawickrama, "Detecting trash and valuables with machine vision in passenger vehicles," Aug. 2020, accepted: 2020-08-23T17:20:37Z. [Online]. Available: https://aaltodoc.aalto.fi:443/handle/123456789/46148

[15] M. Fulton, J. Hong, M. J. Islam, and J. Sattar, "Robotic Detection of Marine Litter Using Deep Visual Detection Models," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 5752–5758, iSSN: 2577-087X.

[16] "TACO: Trash Annotations in Context for Litter Detection," Mar. 2020, arXiv:2003.06975 [cs]. [Online]. Available: http://arxiv.org/abs/2003.06975

[17] S. Zhang, Y. Chen, Z. Yang, and H. Gong, "Computer Vision Based Two-stage Waste Recognition-Retrieval Algorithm for Waste Classification," *Resources, Conservation and Recycling*, vol. 169, p. 105543, June 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0921344921001506

[18] M. Yang and G. Thung, "Classification of Trash for Recyclability Status," p. 6.

[19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[20] F. Chollet and others, "Keras," 2015. [Online]. Available: https://keras.io

[21] "Rich feature hierarchies for accurate object detection and semantic segmentation," Oct. 2014, arXiv:1311.2524 [cs].

[22] "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Jan. 2016, arXiv:1506.01497 [cs]. [Online]. Available: http://arxiv.org/abs/1506.01497

[23] "Mask R-CNN," Jan. 2018, arXiv:1703.06870 [cs]. [Online]. Available: http://arxiv.org/abs/1703.06870

[24] "Going Deeper with Convolutions," Sept. 2014, arXiv:1409.4842 [cs]. [Online]. Available: http://arxiv.org/abs/1409.4842

[25] "Rethinking the Inception Architecture for Computer Vision," Dec. 2015, arXiv:1512.00567 [cs]. [Online]. Available: http://arxiv.org/abs/1512.00567

[26] "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," Aug. 2016, arXiv:1602.07261 [cs]. [Online]. Available: http://arxiv.org/abs/1602.07261

[27] "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," Apr. 2017, arXiv:1704.04861 [cs]. [Online]. Available: http://arxiv.org/abs/1704.04861

[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," 2016, vol. 9905, pp. 21–37, arXiv:1512.02325 [cs]. [Online]. Available: http://arxiv.org/abs/1512.02325

[29] "You Only Look Once: Unified, Real-Time Object Detection," May 2016, arXiv:1506.02640 [cs]. [Online]. Available: http://arxiv.org/abs/1506.02640

[30] "Focal Loss for Dense Object Detection," Feb. 2018, arXiv:1708.02002 [cs]. [Online]. Available: http://arxiv.org/abs/1708.02002

[31] "Waste Bin Detection Dataset (RISS2021)." [Online]. Available: https://www.kaggle.com/datasets/elirotondo/waste-bin-detection-dataset-riss-2021

[32] "Waste Bin Multi-Class Detection Dataset (RISS2021)." [Online]. Available: https://www.kaggle.com/datasets/elirotondo/waste-bin-multiclass-detection-dataset-riss2021

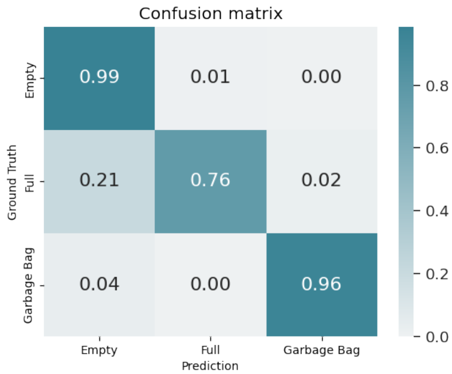
[33] "Microsoft COCO: Common Objects in Context," Feb. 2015, arXiv:1405.0312 [cs]. [Online]. Available: http://arxiv.org/abs/1405.0312

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 248–255, iSSN: 1063-6919.

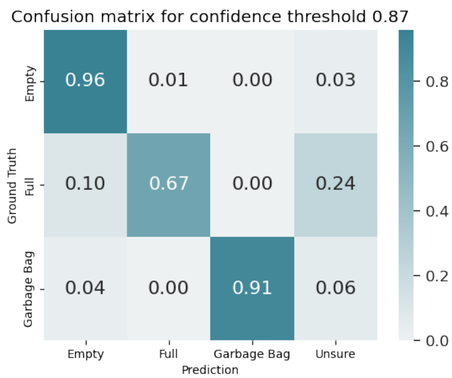
[35] "Feature Pyramid Networks for Object Detection," Apr. 2017, arXiv:1612.03144 [cs]. [Online]. Available: http://arxiv.org/abs/1612.03144

[36] L. Mosley, "A balanced approach to the multi-class imbalance problem," Doctor of Philosophy, Iowa State University, Digital Repository, Ames, 2013, pages: 5050377. [Online]. Available: https://lib.dr.iastate.edu/etd/13537/

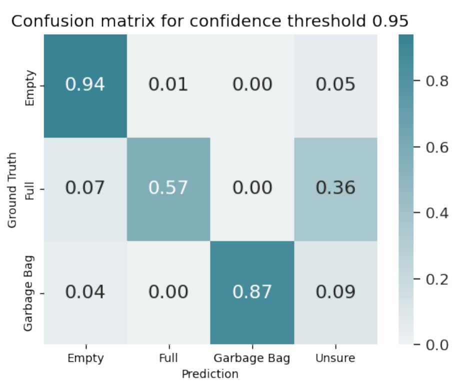
VII. APPENDIX



(a) The proportion of full trash cans falsely identified as empty is more than doubled when applied to the full data set, while the other categories perform similarly.



(b) Applying the same threshold as 6 does not yield nearly as strong of an improvement.



(c) Increasing the threshold even more only marginally improves the rate of misidentifcations but heavily increases the number of ”Unsure” classifications

Fig. 7: Confusion matrices for classification on full test set

Discovery of Heterogeneous Treatment Effects from Landmark Clinical Trials for Cardiovascular Disease

Van Le[1]

Mentors: Chirag Nagpal[2], Artur Dubrawski[2]

***Abstract*— Treatment effects may vary because of the differences in the response characteristics to the interventions of different patients. These characteristics include demographics, phenotypic health characteristics and/or medical history. The presence of treatment effect heterogeneity is, however, often excluded in the evaluation process of Randomized Clinical Trials (RCTs). In this paper, we utilize the existing *Deep Cox Mixtures with Heterogeneous Effects (CMHE)* model [1] to study the heterogeneous treatment effects in three RCTs namely Prevention of Events with Angiotensin Converting Enzyme Inhibition (PEACE) Trial [2], Action to Control Cardiovascular Risk in Diabetes - Lipid Study (ACCORD) [3] and Lipid-Lowering Treatment to Prevent Heart Attack Trial (ALLHAT) [4]. This paper shows that the examined clinical trials were more efficient to certain subgroups of patients with specific phenotypes, thus demonstrating the importance of the discovery of heterogeneous treatment effects. This paper would be beneficial to the stakeholders including doctors and policymakers when deciding on the type of treatment customized for each patient.**

***Index Terms*— Machine Learning for Healthcare, Health Informatics, Personalized Treatment**

I. INTRODUCTION

As of August 2022, there are 423,077 randomized clinical trials (RCTs) conducted worldwide [5]. Each year, the National Institute of Health spends about USD 41.7 billion in medical research [6]. These trials play a vital role in research development of new drugs and implementation of new treatments to patients. However, 400,000 is just a fraction of million trials not registered nor published after concluding that the treatments ineffective. It is possible that the conclusions in the treatment effect in those trials are not considering the differences among patients. If we can show that there are, in fact, two groups of patients which received enhanced/diminished treatment effects from the interventions, we can personalize the treatment and those "ineffective" trials are not just a waste of time and resources.

Each patient in the study sample has different demographic characteristics, baseline physiology, and/or medical history which may affect the treatment outcomes. Despite this heterogeneity, the results shown in many studies implicitly were based on the assumption that the patient's characteristics are the same across the population. It is understandable that the researchers want to give a conclusion at a population-level. However, this may prove to be a double-edged sword when the phenogroups (subgroups of patients based on phenotypes) are not discovered and discussed.

It is in the best interest of patient safety and treatment outcomes to figure out which groups they are in order to guide more precise therapies that result in customized treatment plan. Given that there are more than 400,000 clinical trials available, further research should be done to determine the specific features that make a patient benefit more or less from a special treatment.

To identify subgroups or cohorts of people that show heterogeneous effects to an intervention in the presence of censored outcomes, we use a logical method in this study called Deep Cox Mixtures with Heterogeneous Effects (CMHE). CMHE is a part of the auton-survival, an open-source package for regression, counterfactual estimation, evaluation and phenotyping with censored time-to-event [7]. This model has been verified in its original paper with the evaluation on three datasets discussed in the Related Works section.

This paper also applies some other functions in the auton-survival to plot Kaplan-Meier curves, preprocess and transform data collected at baseline, calculate hazard ratio in the studies, and predict the treatment effects. By fitting the model to the preprocessed data collected at baseline, CMHE will discover the features of the patients that affect the outcome of the interventions and group the patients into two subsets: harmed group benefits more from the control or standard treatment while benefited group benefits more from the experimental or intensive treatment. After that, Decision Tree Classifier will take charge of the classification process where actionable phenotypes are discovered. Thus, when doctors have the baseline characteristics of a new patient, they can utilize the information provided to determine which kind of treatment the patient would most likely benefit from.

In summary, this work applies the CMHE model of the auton-survival package to three large landmark clinical trials that were originally carried out to assess the efficacy of medical interventions to reduce risk of adverse cardiovascular outcomes among patients, and helps with assigning patients to specific treatment group. The CMHE plays the most important role in separating patients into phenogroups before the set of important confounding features are revealed with the application of Decision Tree Classifier models. We propose to reveal the counterfactual phenotypes, thus demonstrating the need of discovering heterogeneous treatment

[1]Van Le studies at Hollins University, Roanoke, VA, USA levh@hollins.edu

[2]Chirag Nagpal and Artur Dubrawski are with the Auton Lab, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA {chiragn,awd}@andrew.cmu.edu

effects in RCTs.

Deep Cox Mixtures with Heterogeneous Effects has been released as part of the open-source package, **auton-survival** and is available at autonlab.github.io/auton-survival/models/cmhe.

II. RELATED WORKS

This section highlights prior works in the discovery of treatment effect heterogeneity. Several approaches for automated assessments of heterogeneous treatment effects have been proposed by different research groups within the machine learning for healthcare field. This section highlights the efficacy of CMHE model which was demonstrated in the original paper by evaluating its performance on two landmark RCTs:

- **Antihypertensive Treatment to Prevent Heart Attack Trial (ALLHAT) [8]**: The Hypertension Study of ALLHAT Trial was to evaluate the performance of a calcium channel blocker or an angiotensin-converting enzyme inhibitor in reducing the incidence of coronary heart disease (CHD) or myocardial infarction (MI) or other cardiovascular disease (CVD) events as compared to treatment with a diuretic. The conclusion in their primary publication is that thiazide-type diuretics, the cheaper treatment, outweigh its competitors in protecting patients from one or more major forms of CVD. Thus, diuretics are recommended for first-step antihypertensive therapy. CMHE, however, identified that older patients with decreased renal function and fewer baseline cardiac diseases will gain more benefits from the chlorthalidone treatment. On the other hand, amlodipine or lisinopril treatment were revealed to be more effective to patients with higher baseline seated blood pressure (systolic and diastolic), lower weight and body mass index, higher baseline kidney function, and are Black and/or non-Hispanic.
- **Action to Control Cardiovascular Risk in Diabetes (ACCORD) - Glycemia Trial [9]**: The Glycemia Trial of ACCORD was to establish the effectiveness of intensive therapy in targeting normal glycated hemoglobin levels, thus decreasing the number of cardiovascular events in patients with type 2 diabetes who had either established CVD or additional cardiovascular risk factors. In the primary publication, it was concluded that as compared with standard therapy, the use of intensive therapy to target normal glycated hemoglobin levels for 3.5 years increased death rate and did not considerably reduce major cardiovascular incidences. Again, with the application of CMHE, it was discovered that enhanced treatment effects could have been seen in patients with worse baseline kidney function (estimated glomerular filtration rate (GFR) $<$ 79.9 mL/min, urine creatinine $<$ 42.9 mg/dL), higher baseline fasted glucose levels, and without a clear history of cerebrovascular disease.

The aforementioned paper also connected their findings with current literature and available information that explain the correlation between the phenotyping process and baseline features of patients. The performance of the model at different time horizons was also evaluated in the form of brier score and time dependent concordance index. This helped reinforce the validity and performance of the examined model.

III. METHODOLOGY

In this paper, we used the **auton-survival** package and its **CMHE** model to extract the two phenogroups with enhanced or diminished treatment effects. After that, we used a Decision Tree Classifier to identify the features that affected the treatment outcomes.

A. Preprocess Data

Data preprocessing is one of the most crucial parts of this project. The accuracy of our model heavily depends on the accurate transformation of raw data into understandable format. All features used in the clinical trials were recorded at baseline when an initial measurement of each patient's conditions including their demographic and clinical characteristics were taken. Features were separated into numerical and categorical features, and preprocessed with a built-in class from **auton-survival** named **Preprocessor**. Because there are missing data in the original datasets, this Preprocessor class is helpful in data imputation, which is the substitution of estimated values for missing or consistentdata items. It also does data scaling to rescale numerical features i.e. normalize their range for the sake of model application. All confounding features are listed in the APPENDIX. For all datasets, we used all features recorded at baseline of the trials.

B. Instantiate and Fit CMHE Model

After preprocessing the feature data, we instantiated the CMHE model with appropriate set of hyperparameters and fitted it using stochastic Expectation Maximization [10].

C. Returns The Estimated Latent Treatment Effect Group ϕ Given The Confounders X

After instantiating the model, we got the estimated latent treatment effect variable ϕ given the confounding features. This latent group ϕ mediated the treatment effect. At this step, each patient is assigned two probabilities of treatment effect corresponding to the two interventions, treatment or control. If the patient has higher probability of event-free survival in the control group, s/he will be assigned to the **Harmed Group**. Otherwise, s/he will be assigned to the **Benefited Group**. After this step, we got the hazard ratio and 95% confidence interval of hazard ratio for each group. The hazard ratio is equivalent to the odds that an individual in the group with the higher hazard reaches the endpoint (i.e. outcome in Table I) first. Thus, in a clinical trial examining time to disease resolution, it represents the odds that a treated patient will resolve symptoms before a control patient. In other words, the higher the hazard ratio, the higher chance of the treated patient's healing first. This pattern is verified in our summary statistics (see Tables II and III).

Dataset	Outcome	Treatment	Control	Hazard Ratio	Event Rate	N
PEACE	Primary End Point	Trandolapril	Placebo	0.96 $\pm$ 0.08	7.97 %	8,290
ALLHAT	CHD Death Plus Nonfatal MI	Pravastatin	Placebo	0.89 $\pm$ 0.09	7.74 %	10,355
ACCORD	Primary End Point	Lipid Fibrate	Lipid Placebo	1.08 $\pm$ 0.18	8.96 %	5,518

TABLE I: Summary statistics of the datasets used in the paper

D. Discover Actionable Phenotypes of Patients

Finally, we applied Decision Tree Classifier models on the data to get the features that affected treatment outcomes. The numerical data used in this step are data before normalization so that it is interpretable. The classifications based on features of patients are shown in the APPENDIX. The depth of each tree is limited to 5 to provide better generalization.

IV. EXPERIMENTS

In our experiments, we consider data from three landmark clinical trials including PEACE, ACCORD Lipid Therapy Trial and ALLHAT Lipid Study originally conducted to determine the optimal treatment for reducing risk from cardiovascular diseases. These datasets were chosen because in their primary publications, no statistically significant difference is shown between the event-free survival probabilities of patients in two groups: treatment versus control. See APPENDIX.

A. Data sets

- **Prevention of Events with Angiotensin Converting Enzyme Inhibition (PEACE) Trial**: The PEACE clinical trial was constituted to establish the appropriate intervention between 4 mg trandolapril per day (treatment) and matching placebo (control) for patients with stable coronary artery disease and normal or slightly reduced left ventricular function. Patients were enrolled over a seven-year period with a median follow up time of 4.8 years. The complete study involved 8,290 participants older than 50 years of age. 4,158 of participants were assigned to the intensive arm, while 4,132 were assigned to the standard arm.
- **Lipid-Lowering Treatment to Prevent Heart Attack Trial (ALLHAT) - Lipid Study**: The ALLHAT Lipid Study was to determine whether pravastatin or usual care is better at reducing all-cause mortality in older, moderately hypercholesterolemic, hypertensive participants with at least one additional CHD risk factor. Patients aged 55 and above were enrolled over a nine-year period with a mean follow up time of 4.8 years. The study involved 10,355 patients, 5,170 of which were randomized to pravastatin while 5,185 received usual care.
- **Action to Control Cardiovascular Risk in Diabetes (ACCORD) - Lipid Therapy Trial**: The ACCORD Lipid Therapy Trial was conducted to determine whether a combination of a statin plus a fibrate or statin monotherapy would protect patients from the risk of CVD. Subjects were those with type 2 diabetes mellitus who were at high risk for CVD. Patients were enrolled over a five-year period with a mean follow up time of 4.7 years. 2,765 of 5,518 patients with type 2 diabetes who were being treated with open-label simvastatin were randomly assigned to receive masked fenofibrate or while the remaining 2,753 received placebo.

B. Figures and Tables

See APPENDIX for additional information.

Dataset	Hazard Ratio	Event Rate	N
PEACE	1.19 $\pm$ 0.16	33.1 %	2,640
ALLHAT	1.19 $\pm$ 0.20	64.3 %	7,076
ACCORD	1.67 $\pm$ 0.39	46.7 %	2,460

TABLE II: Summary Statistics of Harmed Groups in Each Dataset

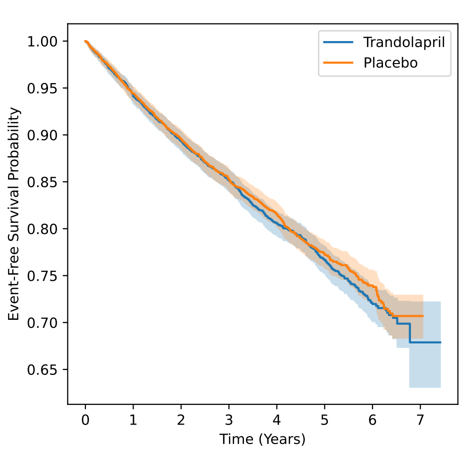
Dataset	Hazard Ratio	Event Rate	N
PEACE	0.86 $\pm$ 0.09	66.9 %	5,650
ALLHAT	0.54 $\pm$ 0.10	35.7 %	3,279
ACCORD	0.73 $\pm$ 0.18	53.3 %	2,818

TABLE III: Summary Statistics of Benefited Groups in Each Dataset

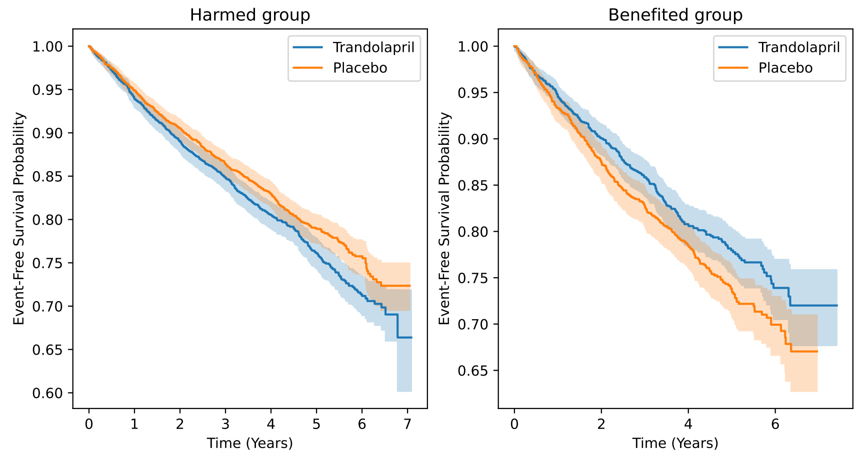
C. Interpretation

- **PEACE Trial**: Among all patients, 5,650 (68.15%) were more likely to benefit from Trandolapril (Hazard Ratio: 0.86 $\pm$ 0.09), demonstrating decreased long-term mortality. On the contrary, 2,640 patients (31.85%) were apparently harmed with the same treatment (HR: 1.19 $\pm$ 0.16) showing increased long-term risk. The classifier revealed patients with a history of smoking and lower SBP and DBP benefit more from Trandolapril.
- **ALLHAT Lipid Study**: Among all patients, 3,279 (31.67%) were more likely to benefit from Pravastatin (Hazard Ratio: 0.54 $\pm$ 0.10), demonstrating decreased long-term mortality. On the contrary, 7,076 patients (68.33%) were apparently harmed with the same treatment (HR: 1.19 $\pm$ 0.20) showing increased long-term risk. Features such as low total cholesterol and high weight would make patients more suitable to be treated with pravastatin while patients with high baseline high-density lipoprotein (HDL) Cholesterol, total cholesterol and lower blood pressure should be given usual care.
- **ACCORD Lipid Therapy Trial**: Among all patients, 2,818 (51.07%) were more likely to benefit from Lipid Fibrate (Hazard Ratio: 0.73 $\pm$ 0.18), demonstrating decreased long-term mortality. On the contrary, 2 patients (48.93%) were apparently harmed with the same

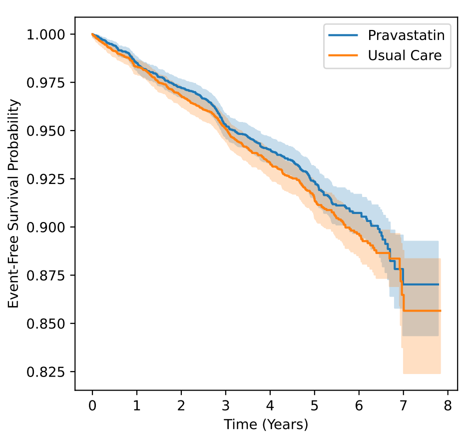
(a) PEACE Original Paper: Cumulative Incidence of the Primary End Point, According to Treatment Group



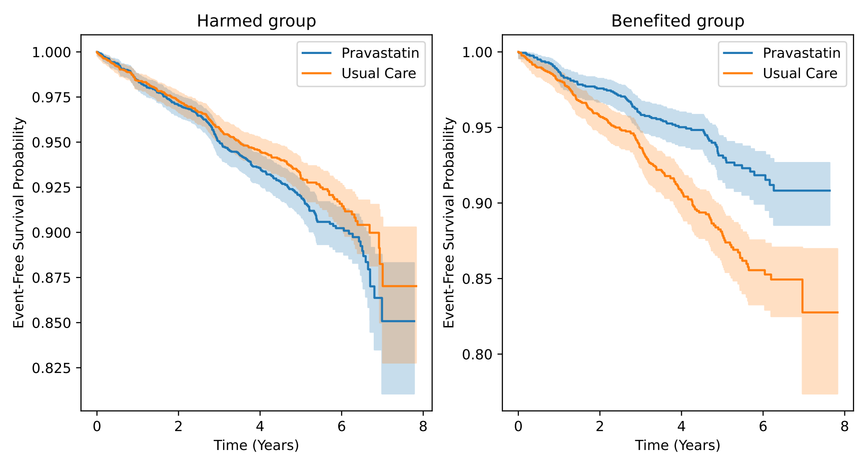
(b) PEACE Phenogroups



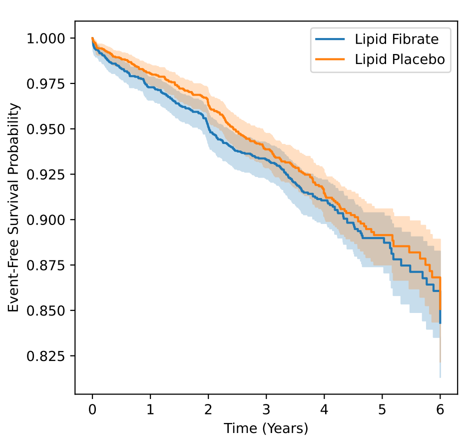
(a) ALLHAT Lipid Study Original Paper: Coronary Heart Disease Death Plus Nonfatal Myocardial Infarction



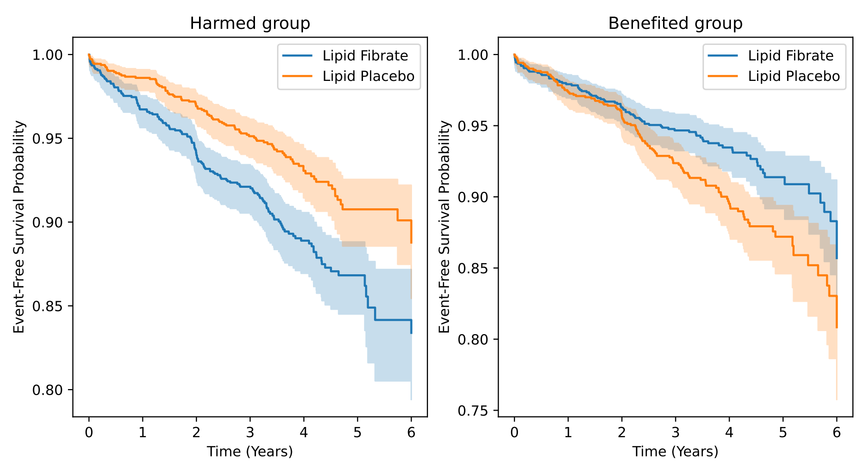
(b) ALLHAT Lipid Phenogroups



(a) ACCORD Lipid Therapy Trial Original Paper: Primary Outcome



(b) ACCORD Lipid Phenogroups



treatment (HR: 1.67 $\pm$ 0.39) showing increased long-term risk. With the application of CMHE and Decision Tree Classifier, several baseline features describing the phenogroup with diminished or enhanced treatment effect from Lipid Fibrate were discovered. Lipid Fibrate seemed to benefit patients who have lower baseline seated diastolic blood pressure, HDL cholesterol and serum creatinine. In contrast, patients with lower baseline glomerular filtration rate, potassium or decreased evidence of cardiovascular disease (lower systolic blood pressure) would receive better protection from cardiovascular events from lipid placebo.

V. LIMITATIONS

For PEACE dataset, we had to use forward selection to select the features that had the most impacts on the treatment effects. The reason is CMHE model was overfitting if it was fitted on all feature data recorded at baseline. For all experiments, we splitted the datasets into train and testing pieces to evaluate the performance of CMHE before running Classifier to discover the phenotypes.

VI. CONCLUSIONS

We applied a novel survival model to different trials and identified a patient subgroup whose outcomes could improve with treatment, even though population level on-average analysis shows no desirable effects. We propose that more research into major clinical trials be conducted to determine the application of the discussed model and provide clinicians with information on how to customize treatment based on each patient's characteristics.

VII. APPENDIX

Additional tables and figures are included on the following page.

ACKNOWLEDGMENT

This material is based upon work supported by Carnegie Mellon University's Center for Machine Learning for Healthcare. Thank you to Carnegie Mellon University, the Auton Lab and Robotics Institute for this research opportunity. A special thanks to the Auton Lab and my mentors—Chirag Nagpal and Dr. Artur Dubrawski—for their mentorship and guidance throughout this research journey. I also want to thank Ms. Rachel Burcin and Dr. John Dolan for their work to help make the RISS program possible.

REFERENCES

[1] C. Nagpal, M. Goswami, K. Dufendach, and A. Dubrawski, "Counterfactual phenotyping with censored time-to-events," *arXiv preprint arXiv:2202.11089*, 2022.

[2] P. T. Investigators, "Angiotensin-converting–enzyme inhibition in stable coronary artery disease," *New England Journal of Medicine*, vol. 351, no. 20, pp. 2058–2068, 2004.

[3] A. S. Group, "Effects of combination lipid therapy in type 2 diabetes mellitus," *New England Journal of Medicine*, vol. 362, no. 17, pp. 1563–1574, 2010.

[4] C. D. Furberg, J. T. Wright, B. R. Davis, J. A. Cutler, M. Alderman, H. Black, W. Cushman, R. Grimm, L. J. Haywood, F. Leenen, *et al.*, "Major outcomes in moderately hypercholesterolemic, hypertensive patients randomized to pravastatin vs usual care: the antihypertensive and lipid-lowering treatment to prevent heart attack trial (allhat-llt)," *Journal of the American Medical Association*, vol. 288, no. 23, pp. 2998–3007, 2002.

[5] "Home - ClinicalTrials.gov," https://clinicaltrials.gov/.

[6] "Budget National Institutes of Health (NIH)," https://www.nih.gov/about-nih/what-we-do/budget.

[7] C. Nagpal, W. Potosnak, and A. Dubrawski, "auton-survival: an open-source package for regression, counterfactual estimation, evaluation and phenotyping with censored time-to-event data," *arXiv preprint arXiv:2204.07276*, 2022.

[8] P. H. A. Trial, "Major outcomes in high-risk hypertensive patients," *Jama*, vol. 288, no. 23, pp. 2981–2997, 2002.

[9] A. to Control Cardiovascular Risk in Diabetes Study Group, "Effects of intensive glucose lowering in type 2 diabetes," *New England journal of medicine*, vol. 358, no. 24, pp. 2545–2559, 2008.

[10] C. Nagpal, S. Yadlowsky, N. Rostamzadeh, and K. Heller, "Deep cox mixtures for survival regression," in *Machine Learning for Healthcare Conference*. PMLR, 2021, pp. 674–708.

[11] C. Nagpal, D. Wei, B. Vinzamuri, M. Shekhar, S. E. Berger, S. Das, and K. R. Varshney, "Interpretable subgroup discovery in treatment effect estimation with application to opioid prescribing guidelines," in *Proceedings of the ACM Conference on Health, Inference, and Learning*, 2020, pp. 19–29.

Name	Mean	Standard Error
Seated Systolic Blood Pressure	135.10	0.32
Seated Diastolic Blood Pressure	78.77	0.18

TABLE IV: Summary Statistics of Numerical Features of Patients in Harmed Groups in PEACE Trial at Baseline

Name	Mean	Standard Error
Seated Systolic Blood Pressure	123.92	0.20
Seated Diastolic Blood Pressure	72.77	0.13

TABLE V: Summary Statistics of Numerical Features of Patients in Benefited Groups in PEACE Trial at Baseline

Name	Mean	Standard Error
Seated Systolic Blood Pressure	142.76	0.16
Seated Diastolic Blood Pressure	83.95	0.11
Age	66.33	0.09
Total Cholesterol	222.04	0.32
Glucose	109.64	0.35
Weight (LBS)	186.23	0.47
Height (INC)	66.51	0.05
HDL Cholesterol	46.28	0.15
LDL Cholesterol	145.55	0.26
Triglycerides	146.91	0.77
BMI	29.69	0.67

TABLE VI: Summary Statistics of Numerical Features of Patients in Harmed Groups in ALLHAT Trial at Baseline

Name	Mean	Standard Error
Seated Systolic Blood Pressure	149.74	0.22
Seated Diastolic Blood Pressure	84.09	0.18
Age	66.48	0.13
Total Cholesterol	227.32	0.46
Glucose	161.24	1.35
Weight (LBS)	176.49	0.65
Height (INC)	64.61	0.06
HDL Cholesterol	50.15	0.27
LDL Cholesterol	145.45	0.36
Triglycerides	162.59	1.48
BMI	29.85	0.10

TABLE VII: Summary Statistics of Numerical Features of Patients in Benefited Groups in ALLHAT Trial at Baseline

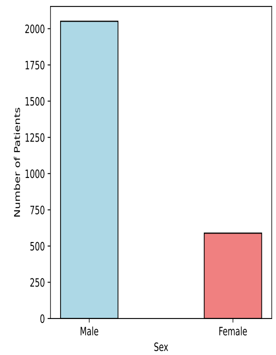
Name	Mean	Standard Error
Seated Systolic Blood Pressure	134.05	0.35
Seated Diastolic Blood Pressure	71.99	0.20
Age	64.39	0.13
Total Cholesterol	183.18	0.78
Heart Rate	69.86	0.22
Potassium	4.46	0.01
Fasting Plasma Glucose	162.32	0.95
HDL Cholesterol	37.92	0.16
LDL Cholesterol	104.08	0.64
Triglycerides	212.58	2.35
VLDL Cholesterol	40.99	0.40
Alanine Transaminase	28.00	0.29
Creatine Phosphokinase	133.00	2.05
Serum Creatinine	0.93	0.00
Estimated Glomerular Filtration Rate	87.46	0.47
Urine Albumine	14.99	0.75
Urine Creatinine	1.07	0.01
Urine Albumin/Creatinine Ratio	15.35	0.75

TABLE VIII: Summary Statistics of Numerical Features of Patients in Harmed Groups in ACCORD Trial at Baseline

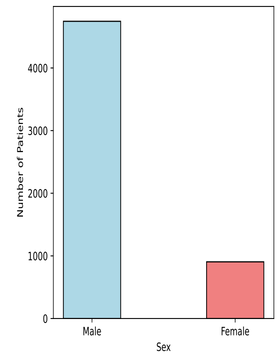
Name	Mean	Standard Error
Seated Systolic Blood Pressure	133.25	0.32
Seated Diastolic Blood Pressure	75.33	0.19
Age	61.35	0.12
Total Cholesterol	167.82	0.63
Heart Rate	74.31	0.22
Potassium	4.48	0.01
Fasting Plasma Glucose	185.91	1.00
HDL Cholesterol	38.31	0.14
LDL Cholesterol	97.31	0.54
Triglycerides	162.53	1.57
VLDL Cholesterol	32.16	0.30
Alanine Transaminase	27.04	0.23
Creatine Phosphokinase	139.31	2.08
Serum Creatinine	0.92	0.00
Estimated Glomerular Filtration Rate	91.95	0.39
Urine Albumine	4.94	0.20
Urine Creatinine	1.40	0.01
Urine Albumin/Creatinine Ratio	3.96	0.17

TABLE IX: Summary Statistics of Numerical Features of Patients in Benefited Groups in ACCORD Trial at Baseline

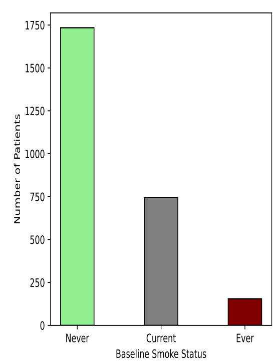
(a) Distribution of Patients in Harmed Group by Sex in PEACE



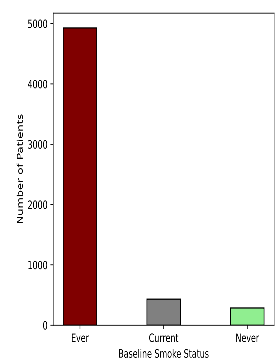
(b) Distribution of Patients in Benefited Group by Sex in PEACE



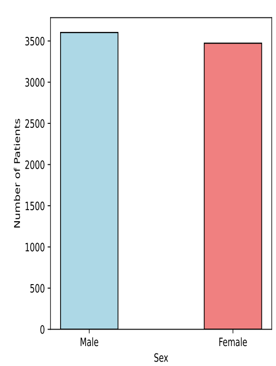
(a) Distribution of Patients in Harmed Group by Smoke Status at Baseline in PEACE



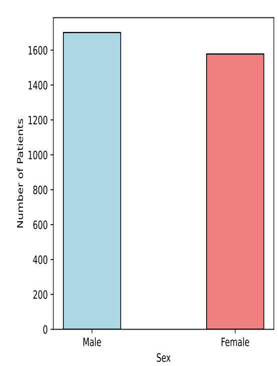
(b) Distribution of Patients in Benefited Group by Smoke Status at Baseline in PEACE



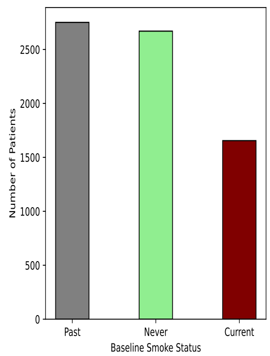
(a) Distribution of Patients in Harmed Group by Sex in ALLHAT



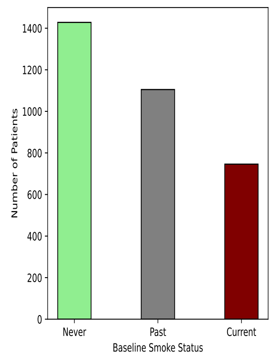
(b) Distribution of Patients in Benefited Group by Sex in ALLHAT



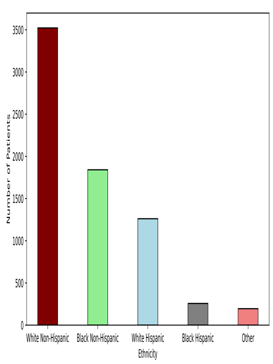
(a) Distribution of Patients in Harmed Group by Smoke Status at Baseline in ALLHAT



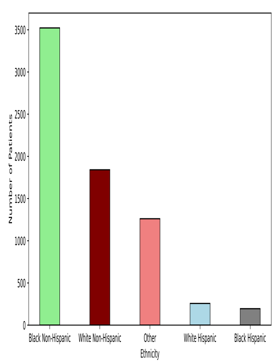
(b) Distribution of Patients in Benefited Group by Smoke Status at Baseline in ALLHAT



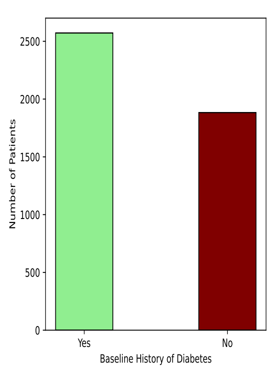
(a) Distribution of Patients in Harmed Group by Ethnicity in ALLHAT



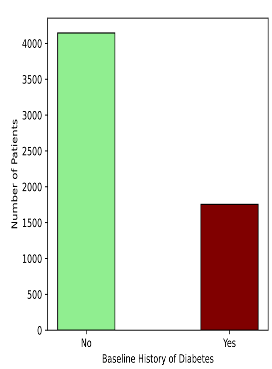
(b) Distribution of Patients in Benefited Group by Ethnicity in ALLHAT



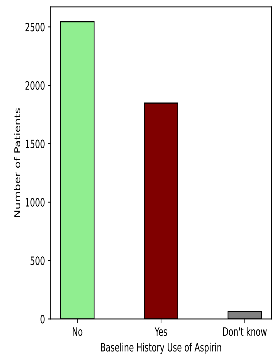
(a) Distribution of Patients in Harmed Group by History of Diabetes in ALLHAT



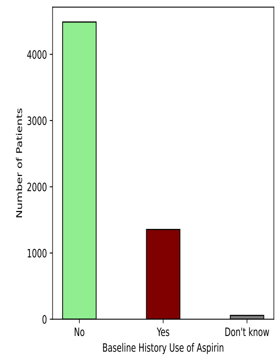
(b) Distribution of Patients in Benefited Group by History of Diabetes in ALLHAT



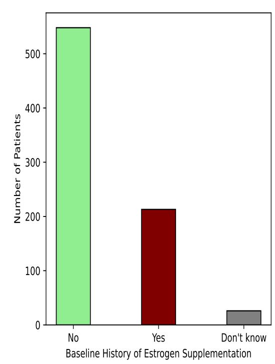
(a) Distribution of Patients in Harmed Group by History Use of Aspirin in ALLHAT



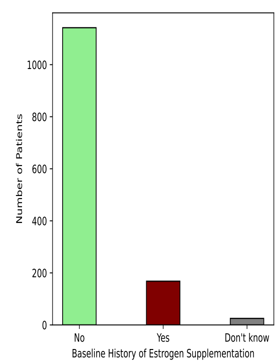
(b) Distribution of Patients in Benefited Group by History Use of Aspirin in ALLHAT



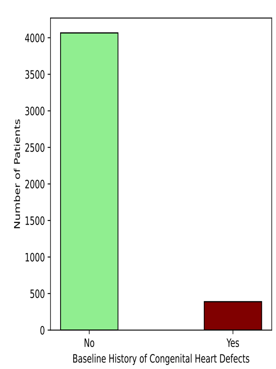
(a) Distribution of Patients in Harmed Group by History of Estrogen Supplementation in ALLHAT



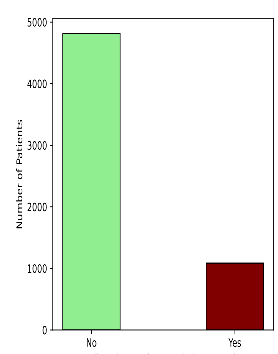
(b) Distribution of Patients in Benefited Group by History of Estrogen Supplementation in ALLHAT

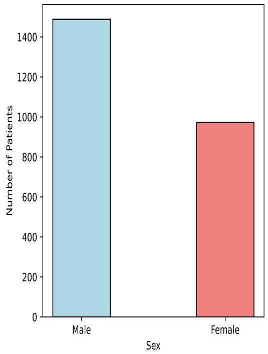


(a) Distribution of Patients in Harmed Group by History of Congenital Heart Defects in ALLHAT

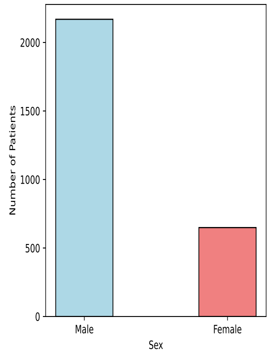


(b) Distribution of Patients in Benefited Group by History of Congenital Heart Defects in ALLHAT

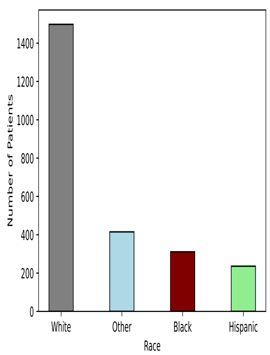




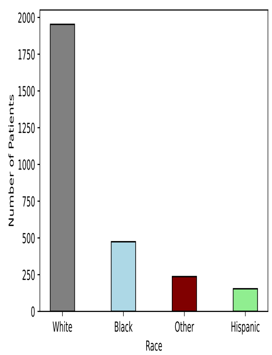
(a) Distribution of Patients in Harmed Group by Sex in ACCORD



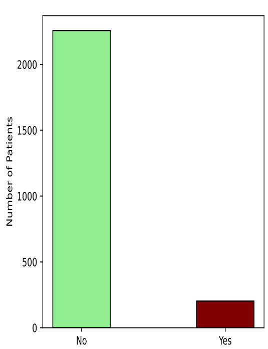
(b) Distribution of Patients in Benefited Group by Sex in ACCORD



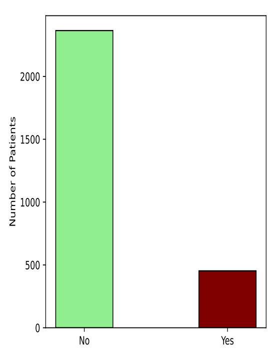
(a) Distribution of Patients in Harmed Group by Race in ACCORD



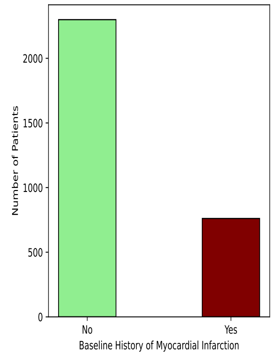
(b) Distribution of Patients in Benefited Group by Race in ACCORD



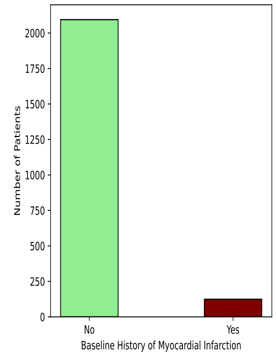
(a) Distribution of Patients in Harmed Group by Baseline Smoke Status in ACCORD



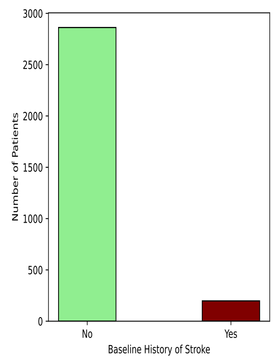
(b) Distribution of Patients in Benefited Group by Baseline Smoke Status in ACCORD



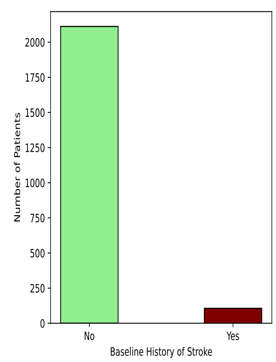
(a) Distribution of Patients in Harmed Group by Baseline History of Myocardial Infarction in ACCORD



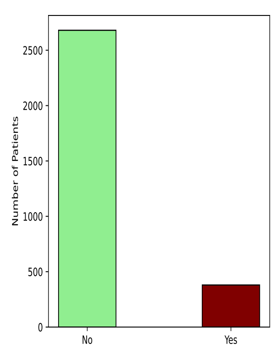
(b) Distribution of Patients in Benefited Group by Baseline History of Myocardial Infarction in ACCORD



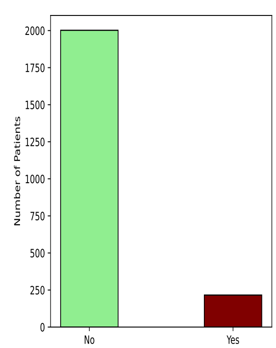
(a) Distribution of Patients in Harmed Group by Baseline History of Stroke in ACCORD



(b) Distribution of Patients in Benefited Group by Baseline History of Stroke in ACCORD

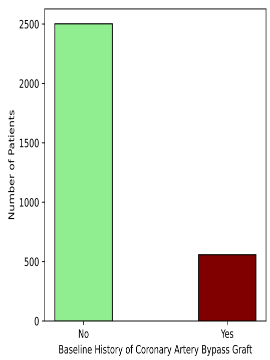


(a) Distribution of Patients in Harmed Group by Baseline History of Angina in ACCORD

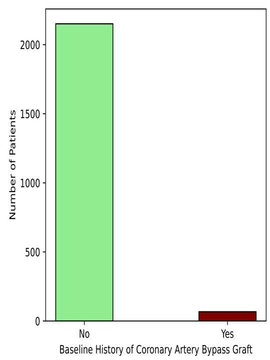


(b) Distribution of Patients in Benefited Group by Baseline History of Angina in ACCORD

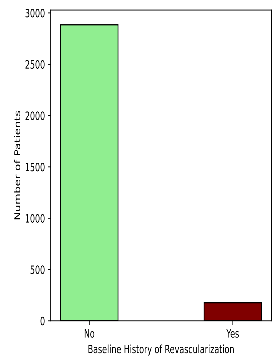
(a) Distribution of Patients in Harmed Group by Baseline History of Coronary Artery Bypass Graft in ACCORD



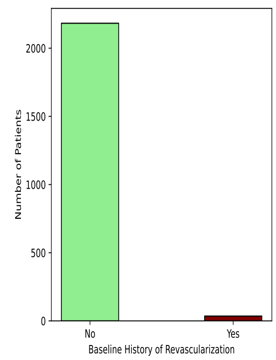
(b) Distribution of Patients in Benefited Group by Baseline History of Coronary Artery Bypass Graft in ACCORD



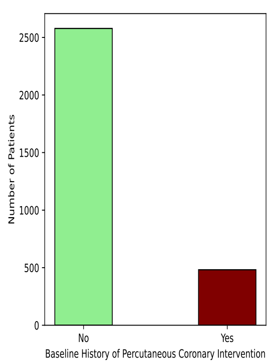
(a) Distribution of Patients in Harmed Group by Baseline History of Revascularization in ACCORD



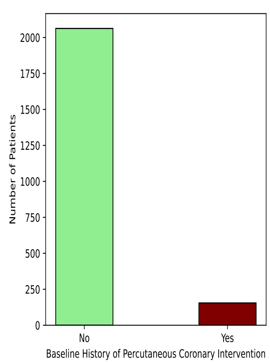
(b) Distribution of Patients in Benefited Group by Baseline History of Revascularization in ACCORD



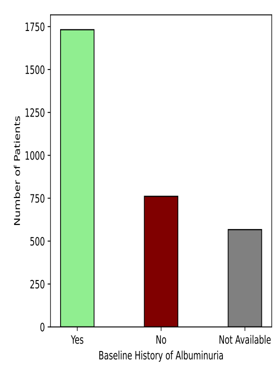
(a) Distribution of Patients in Harmed Group by Baseline History of Percutaneous Coronary Intervention in ACCORD



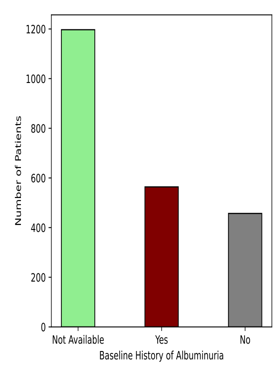
(b) Distribution of Patients in Benefited Group by Baseline History of Percutaneous Coronary Intervention in ACCORD



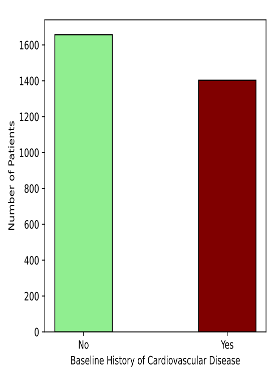
(a) Distribution of Patients in Harmed Group by Baseline History of Albuminuria in ACCORD



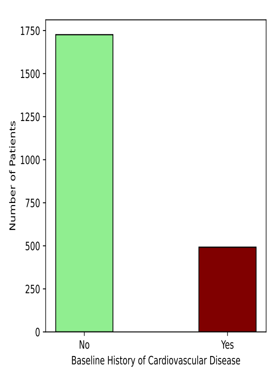
(b) Distribution of Patients in Benefited Group by Baseline History of Albuminuria in ACCORD



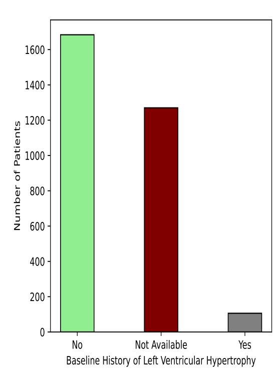
(a) Distribution of Patients in Harmed Group by Baseline History of Cardiovascular Disease in ACCORD



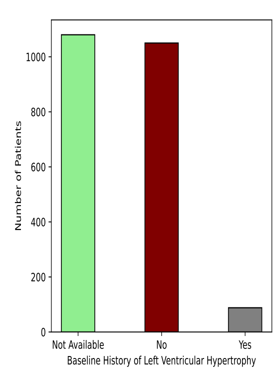
(b) Distribution of Patients in Benefited Group by Baseline History of Cardiovascular Disease in ACCORD



(a) Distribution of Patients in Harmed Group by Baseline History of Left Ventricular Hypertrophy in ACCORD



(b) Distribution of Patients in Benefited Group by Baseline History of Left Ventricular Hypertrophy in ACCORD



Sketch-to-image synthesis using semantic priors

Vihaan Misra[1], Peter Schaldenbrand[2] and Jean Oh[3]

Abstract— **Recent developments in text-to-image synthesis have shown some substantial and intriguing progress for generating images with simply a natural language input. While these methods produce seemingly relevant as well as high-quality images, a clear divide can be seen in terms of giving the user more control over what they actually want to see. Sketches, can communicate visual data in a more natural form than text especially relating to the composition of an image and concept design. Current sketch-to-image translation methods lack generalizability, control over the style of the image, and content control via additional language input. Through this work, we propose a method for rewriting pre-trained General Adversarial Network(GAN) models by incorporating sketch and text inputs from the user. Furthermore, we use a cross-domain adversarial loss paired with Contrastive Language-Image Pre-Training(CLIP) loss to ensure that the generated images match the sketch and text input given by the user whilst retaining good visual quality. This approach also reduces the need for training computationally expensive models through a form of remodelling of off-the-shelf GAN models. We will evaluate our method through user studies to show that employing a sketch and text input pair considerably increases the appropriateness of the generated images and gives the user more supervision over the resulting output.**

Index Terms— **Deep Learning Methods, Visual Learning**

I. INTRODUCTION

“The drawing shows me at one glance what might be spread over ten pages in a book” - Ivan Turgenev [1]

It is commonly known that an image can express information more efficiently than spoken or written language as seen in common expressions such as ”A picture is worth a thousand words” and ”A poet would be overcome by sleep and hunger before being able to describe with words what a painter is able to depict in an instant” ([2], [3]). While only a few words forming a description may lack enough detail to completely describe an image, it has become a common task in Machine Learning to translate natural language descriptions in to images ([4]–[7]). This task has been enabled by large-scale text and image paired datasets [8] leading to models which can robustly encode visual and language information into the same latent representation [9]. While the quality and visual-language consistency of these models has improved vastly in recent years, such new models still lack a number of essential components to generating visual content accurately:

1) **Controlling the output:** These models generate images entirely from text inputs. This means that if the user wishes to make changes, they can only do so either by altering their text input or rephrasing it entirely commonly referred to as Prompt Engineering. While making broader changes pertaining to style and color can be done easily through text, making finer changes like poses, structure and arrangement is challenging. This restricts the user’s ability to make graphic changes to the produced output as it is governed by their skills to articulate the desired changes and the model’s ability to understand natural language cues.
2) **Scalibility:** Such text-to-image models can only provide a small number of outputs. A user creating artwork might want a set of such categories of images. For example, say the user wants a collection of images of elephants in specific situations like with its trunk high up, sitting or facing right. To do so, they would either need to textually particularize the details or produce a sizable dataset and train a generative network on it. Along with gathering the dataset, employing the generative approach also necessitates spending a significant amount of computational power and resources for the model’s training.
3) **Generalizability:** Large and unstructured datasets have a strong negative impact on approaches that use generative models. These models perform admirably in terms of photorealism on carefully curated datsets, particularly when it comes to human faces. But in highly diverse datasets like the ImageNet [10], their performance is proven to be sub-par [11]. As a result, it restrains the current approaches to function with certain types of images, such as faces, cats, and horses, and a new model must be trained specifically for that class.

In this paper, we propose a novel architecture that tackles the aforementioned drawbacks and improves on existing baselines in the task of user-guided image synthesis. We use a textual prompt and hand-drawn sketches from the user as inputs to create a generative model that can create natural and suitable images. To achieve this, we use a StyleGAN-XL [12] based model, guided by Contrastive Language-Image Pre-Training(CLIP) [9] and Learned Perceptual Image Patch Similarity(LPIPS) [13] based loss to create customized Generative Adversarial network(GAN) [14] models that match the user inputs.

The significance of hand-drawn sketches in directing image synthesis has been emphasized ever since Shi-Min Hu’s

[1]Vihaan Misra is with the Department of Electrical Engineering, Netaji Subhas University of Technology, New Delhi, India vihaan.ee19@nsut.ac.in

[2]Peter Schaldenbrand and Jean Oh are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA pschalde, hyaejino@andrew.cmu.edu

Sketch2Photo [15]. Sketches are an attractive form of control because they are simple and abstract enough that nearly anyone can generate and edit one, but equivalent actions in the picture domain require a higher level of creative ability and/or expensive training.

There are seven elements of visual art: space, shapes, lines, colors, form, texture, and value([16], [17]). Most literature on affecting the style of images, such as neural style transfer [18] can control color and textures well but lack changes to compositional elements such as space and shape. State-of-the-art text-to-image synthesis models can control the colors and textures of the image via the language prompt, but it is challenging or impossible to describe compositional information via language. This is why a sketch input is necessary for a model to properly generate art with control over all 7 elements of art.

Moreover, to further facilitate intuitive text-based semantic image manipulation and maintain human perception of image quality, we use the CLIP and LPIPS guided loss functions. We use this method to create modified generative models that can produce new samples as well as interpolate between the generated images. Our method gives extensive control to the user over the output of the model to obtain the desired image. We further carry out experiments and user-studies to establish the validity of our contributions.

II. Related Work

A. Text-based Image Synthesis

Recent advancements in text-to-image synthesis have shown to produce very high resolution images. Generative approaches ([19]–[22]) mainly follow the use of Generators and Discriminators which are trained together while competing with one another. The generator learns to produce a target output whilst the discriminator learns to distinguish true data from the output of the generator. Further, conditional GANs([23]–[26]) are also used for text to image generation. These models produced high-quality outputs on select classes e.g. cats and flowers. Other methods([27]–[29]) use a Vector Quantized (VQ) variational autoencoders to compress the picture into a low-dimensional discrete latent space and match the density of hidden variables. Auto Regressive models [30] treat an image as a sequence of pixels and have been used for image generation([27], [31], [32]). With powerful transformer networks and extensive text-image pair datasets, methods like([4], [33], [34]) have made significant progress in image relevancy and quality in the text-to-image synthesis.

B. Sketch-based Image Retrieval and Synthesis(SBIR)

Utilizing hand-drawn drawings to obtain pictures has been a prevalent field of study. Although Sketch Based-Image Retrieval(SBIR)([35]–[38]) was originally developed for a category-level environment([39]–[41]), it has recently undergone a fine-grained transition to better capture the intrinsic fine-grained qualities of drawings([42]–[44]). Recent SBIR networks have followed the Deep Neural Network approach([35]–[37], [39], [41], [45]). Moreover, sketch-image pair datasets([46]–[51]) have facilitated the creation of generative networks that translate sketches to image outputs([46], [52], [53]), along with several sketch-based editing models([54], [55]).

C. Model Fine-Tuning

Recent studies use Transfer Learning to fine-tune the weights of a pretrained generator and discriminator pair in order to train a GAN network on a new dataset([10], [56]). Furthermore, in order to prevent overfitting, several approaches involving like limiting the weight changes of the network([57]–[62]) and data augmentation([63]–[66]) were used. We follow a very similar approach to Sketch Your Own GAN [67], wherein we create a customized GAN model for given sketch inputs from the user. We further improve on their method in terms of using a StyleGAN-XL generator instead of a StyleGAN-2 that is pretrained on the ImageNet dataset. This increases the model's generalizability and eliminates the requirement for pretrained StyleGAN models for each desired class. In addition, we provide semantics-aware losses to steer text-based adjustments and maintain high-resolution image quality(explained in later sections).

III. Method

Our approach is an optimization problem in which a generative neural network is optimized to satisfy three criteria. The images that the generator produces must (1) fit the sketch given as input, (2) look photographic, and (3) must match the text description of the image. The parameter space is the fine-tuning of a pretrained generator model. The generator model must be capable of producing images described in the text description and depicted in the sketch. For this reason, we use the StyleGAN-XL model which has been trained on ImageNet dataset which contains 1000 different categories of image contents.

A. Sketch Control

In order to alter the generator model to produce images that fit the sketch, we utilize a pretrained image-to-sketch [68] model following methodology from [67]. In each iteration, the generator produces a batch of images. These generated images can be converted into sketches then compared to the input sketch. We utilize a discriminator model for comparing the input sketches and the sketches of the generated images.

B. Language Control

Images produced by the generator are encoded using a pretrained image-text encoder, CLIP. These encodings are compared to the encodings of the input text prompt using cosine distance. This forms a loss value, which when decreased, ensures that the generated images are similar to the text description.

Prior to encoding the generated images with CLIP, the images are augmented with random perspective shift, cropping, and color normalization. This step is necessary for acquiring a robust loss signal that can be back-propagated through the

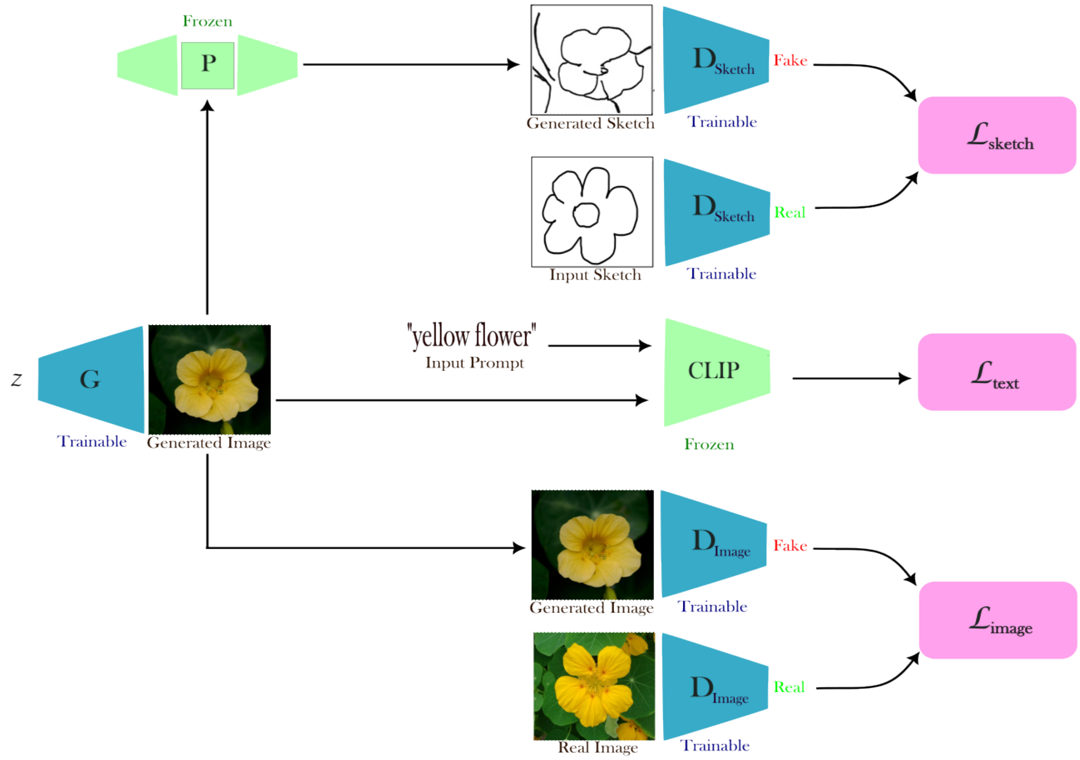


Fig. 1. Training procedure. Our training consists of three major components. (a) L_{sketch}: the sketch discriminator classifies between fake and user sketches. A pretrained mapping network P [68]is used to translate the output of our model G to a fake sketch. (b) L_{text}: the text-guiding power of CLIP [9] is leveraged for calculating a semantics-based loss. (c) L_{image}: the image discriminator D_x classifies between fake and real images. Real images are sampled from the training set of the original model G.

pretrained CLIP model and is common in other text-to-image methods ([6], [7], [69]).

C. Image Quality

To ensure that the images produced by the generator are photographic in appearance, we draw from traditional image GAN literature. The generated images are compared to a data set of real photographs using a discriminator model.

To bridge the gap between sketch training data and the image generative model, we introduce a cross-domain adversarial loss to encourage the generated images to match the sketches Y. Before passing into the discriminator, the output of the generator is transferred into a sketch by the pretrained image-to-sketch network F

Optimization Equation

$$\mathcal{L} = \mathcal{L}_{\text{sketch}} + \lambda_{\text{image}} \mathcal{L}_{\text{text}} + \lambda_{\text{text}} \mathcal{L}_{\text{text}}$$

with λ_{image} controlling the importance of the image regularization term and λ_{text} controlling the semantics term. These regularization terms decide the importance being given to the user sketch and text inputs. The user can alter the value of these terms in order to decide which input to give more weightage to in accordance with the image they want to create.

We aim to learn a new set of weights $G(\mathbf{z}; \theta')$, with the following minimax objective:

$$\theta' = \arg\min_{\theta'} \max_{D_X, D_Y} \mathcal{L}$$

IV. Results

We evaluate our method's performance using human-based perception metrics and numerical metrics. We also demonstrate new creative capabilities enabled by the flexible controllability of our approach. We utilise PhotoSketch [68] to convert the cat photos from the LSUN dataset [70], then manually choose groups of 30 sketches with comparable forms and positions to serve as the user input, as illustrated in Figure 2. We manually select an additional 2,500 photos that correspond to the input sketches to determine the target distribution. Out of 10,000 candidate photos, we choose the ones with the closest chamfer distances [56] to the specified inputs. Only the 10 chosen sketches are available to the model; the sets of 2,500 genuine photos serve as representations of the actual but unseen target distributions.

V. Discussion

We propose a new approach to the sketch-to-image as well as the text-to-image domains, which have recently seen a variety of novel methods aimed at increasing the image quality and relevancy to the input text. While some of the methods do incorporate image editing techniques, they still lack a sense of giving the user a more generalized control over the output of the model. Our method overcomes

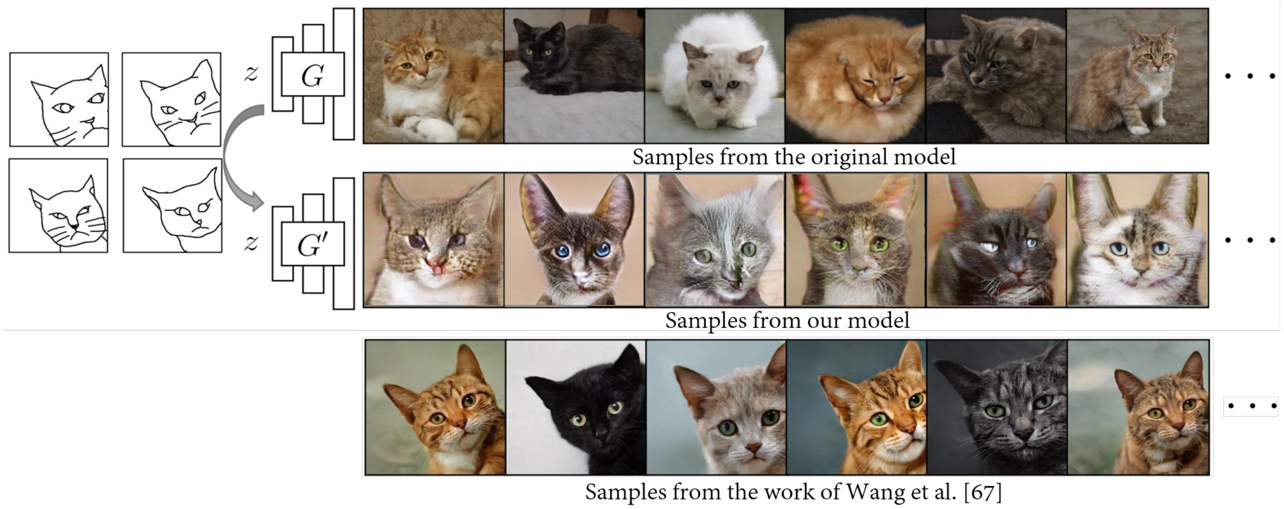


Fig. 2. Each row shows uncurated samples generated from a model trained on sample sketches from Quickdraw [71]. Same noise z is used and truncation ψ = 0.5 is applied to each model.

the domain gaps between sketches, texts and the generator parameter space to create a pipeline that can synthesize an infinite number of unique images using just one or more hand-drawn sketches. Additionally, the user can select the regularisation terms to determine the significance of the sketch or text input. This provides the users the freedom to choose if they want the sketch to be given more weight than the written description or vice-versa. Thus, our approach of using off-the-shelf GAN models and giving the user more control over the training process by using text and sketch inputs leads to a higher favorability of our approach in human evaluations and objective metrics. The results in Figure 2 show that the StyleGAN-XL produces lower quality images as compared to [67]. The advantage of using the StyleGAN-XL generator, however, is that it can be used to create edited images across all the categories of Imagenet. This increases the generalizability of our approach because it means that it is not constrained to the smaller categories that existing GAN models are trained on.

However, plenty of room for improvement remains for our method. Significant changes introduced in StyleGAN-3 and StyleGAN-XL architectures pose many challenges. Central among these is the disentanglement of its latent spaces and the ability to accurately invert and edit real images. Figuring out a better algorithm to traverse this latent space can hugely improve the quality of our outputs. Another drawback of our current approach is that since our models require more than 30K training iterations, tweaking a model in real-time is not achievable. Moreover, because our approach needs access to the original model's training data, it might not be suited for situations where the training data is not readily available.

VI. Future Work

Acknowledgment

Sincere thanks go out to Dr. Jean Oh and Peter Schaldenbrand for supervising this summer's study, as well as for their crucial mentorship and direction. We are grateful to Dr. John Dolan and Rachel Burcin for organizing RISS. Finally, would like to express our gratitude to the CMU Robotics Institute for supporting this study.

References

[1] I. S. Turgenev, 1818-1883, *Fathers and sons.* Oxford ; New York : Oxford University Press, 2008. [Online]. Available: https://search.library.wisc.edu/catalog/9910097537202121

[2] J. Janson, A. F., and H. W., *History of art.* Prentice-Hall ; Abrams.

[3] M.-M. Fourcade, *L'arche de Noé: réseau Alliance, 1940 - 1945*, ser. Histoire contemporaine. Plon.

[4] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," 2022. [Online]. Available: https://arxiv.org/abs/2204.06125

[5] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi, "Photorealistic text-to-image diffusion models with deep language understanding." [Online]. Available: http://arxiv.org/abs/2205.11487

[6] K. Frans, L. B. Soros, and O. Witkowski, "Clipdraw: Exploring text-to-drawing synthesis through language-image encoders," *CoRR*, vol. abs/2106.14843, 2021. [Online]. Available: https://arxiv.org/abs/2106.14843

[7] P. Schaldenbrand, Z. Liu, and J. Oh, "Styleclipdraw: Coupling content and style in text-to-drawing translation," 2022. [Online]. Available: https://arxiv.org/abs/2202.12362

[8] C. Schuhmann, R. Vencu, R. Beaumont, R. Kaczmarczyk, C. Mullis, A. Katta, T. Coombes, J. Jitsev, and A. Komatsuzaki, "LAION-400m: Open dataset of CLIP-filtered 400 million image-text pairs." [Online]. Available: http://arxiv.org/abs/2111.02114

[9] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision." [Online]. Available: http://arxiv.org/abs/2103.00020

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, pp. 248–255. [Online]. Available: https://ieeexplore.ieee.org/document/5206848/

[11] T. Grigoryev, A. Voynov, and A. Babenko, "When, why, and which pretrained GANs are useful?" [Online]. Available: http://arxiv.org/abs/2202.08937

[12] A. Sauer, K. Schwarz, and A. Geiger, "StyleGAN-XL: Scaling StyleGAN to large diverse datasets." [Online]. Available: http://arxiv.org/abs/2202.00273
[13] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," 2018. [Online]. Available: https://arxiv.org/abs/1801.03924
[14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014. [Online]. Available: https://arxiv.org/abs/1406.2661
[15] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu, "Sketch2photo: Internet image montage," vol. 28, no. 5, pp. 1–10, place: New York, NY, USA Publisher: Association for Computing Machinery. [Online]. Available: https://doi.org/10.1145/1618452.1618470
[16] P. D. Beck, "Fourth-grade students' subjective interactions with the seven elements of art: An exploratory case study using q-methodology," ISBN: 978-1-321-39774-1 Publication Title: ProQuest Dissertations and Theses 3666682. [Online]. Available: https://www.proquest.com/dissertations-theses/fourth-grade-students-subjective-interactions/docview/1640918179/se-2
[17] M. Thorson, "Formal elements of art," *Encounters With the Arts: Readings for ARTC150*, 2020.
[18] N. Kolkin, J. Salavon, and G. Shakhnarovich, "Style transfer by relaxed optimal transport and self-similarity," 06 2019, pp. 10 043–10 052.
[19] M. Cha, Y. L. Gwon, and H. T. Kung, "Adversarial learning of semantic relevance in text to image synthesis," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 3272–3279, Jul. 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4553
[20] J. Cheng, F. Wu, Y. Tian, L. Wang, and D. Tao, "Rifegan: Rich feature generation for text-to-image synthesis from prior knowledge," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
[21] H.-Y. Tseng, L. Jiang, C. Liu, M.-H. Yang, and W. Yang, "Regularizing generative adversarial networks under limited data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 7921–7931.
[22] L. Gao, D. Chen, J. Song, X. Xu, D. Zhang, and H. T. Shen, "Perceptual pyramid adversarial networks for text-to-image synthesis," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 8312–8319, Jul. 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4844
[23] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1060–1069. [Online]. Available: https://proceedings.mlr.press/v48/reed16.html
[24] J. Liang, W. Pei, and F. Lu, "Cpgan: Full-spectrum content-parsing generative adversarial networks for text-to-image synthesis," 2019. [Online]. Available: https://arxiv.org/abs/1912.08562
[25] T. Qiao, J. Zhang, D. Xu, and D. Tao, "Mirrorgan: Learning text-to-image generation by redescription," 2019. [Online]. Available: https://arxiv.org/abs/1903.05854
[26] M. Zhu, P. Pan, W. Chen, and Y. Yang, "Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis," 2019. [Online]. Available: https://arxiv.org/abs/1904.01310
[27] A. v. d. Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," 2017. [Online]. Available: https://arxiv.org/abs/1711.00937
[28] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with vq-vae-2," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Paper.pdf
[29] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 12 873–12 883.
[30] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020. [Online]. Available: https://arxiv.org/abs/2005.14165
[31] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, "Generative pretraining from pixels," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 1691–1703. [Online]. Available: https://proceedings.mlr.press/v119/chen20s.html
[32] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 4055–4064. [Online]. Available: https://proceedings.mlr.press/v80/parmar18a.html
[33] M. Ding, Z. Yang, W. Hong, W. Zheng, C. Zhou, D. Yin, J. Lin, X. Zou, Z. Shao, H. Yang, and J. Tang, "Cogview: Mastering text-to-image generation via transformers," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=cnWSyJNmeCE
[34] J. Lin, R. Men, A. Yang, C. Zhou, M. Ding, Y. Zhang, P. Wang, A. Wang, L. Jiang, X. Jia, J. Zhang, J. Zhang, X. Zou, Z. Li, X. Deng, J. Liu, J. Xue, H. Zhou, J. Ma, J. Yu, Y. Li, W. Lin, J. Zhou, J. Tang, and H. Yang, "M6: A Chinese Multimodal Pretrainer," *arXiv e-prints*, p. arXiv:2103.00823, Mar. 2021.
[35] J. Collomosse, T. Bui, and H. Jin, "Livesketch: Query perturbations for guided sketch-based visual search," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2874–2882.
[36] S. Dey, P. Riba, A. Dutta, J. Lladós, and Y. Song, "Doodle to search: Practical zero-shot sketch-based image retrieval," *CoRR*, vol. abs/1904.03451, 2019. [Online]. Available: http://arxiv.org/abs/1904.03451
[37] A. Dutta and Z. Akata, "Semantically tied paired cycle consistency for zero-shot sketch-based image retrieval," *CoRR*, vol. abs/1903.03372, 2019. [Online]. Available: http://arxiv.org/abs/1903.03372
[38] A. Sain, A. K. Bhunia, V. Potlapalli, P. N. Chowdhury, T. Xiang, and Y.-Z. Song, "Sketch3t: Test-time training for zero-shot sbir," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 7462–7471.
[39] L. S. F. Ribeiro, T. Bui, J. Collomosse, and M. Ponti, "Sketchformer: Transformer-based representation for sketched structure," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
[40] Y. Shen, L. Liu, F. Shen, and L. Shao, "Zero-shot sketch-image hashing," *CoRR*, vol. abs/1803.02284, 2018. [Online]. Available: http://arxiv.org/abs/1803.02284
[41] J. Zhang, F. Shen, L. Liu, F. Zhu, M. Yu, L. Shao, H. T. Shen, and L. Van Gool, "Generative domain-migration hashing for sketch-to-image retrieval," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
[42] Q. Yu, F. Liu, Y.-Z. Song, T. Xiang, T. M. Hospedales, and C. C. Loy, "Sketch me that shoe," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 799–807.
[43] J. Song, Q. Yu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Deep spatial-semantic attention for fine-grained sketch-based image retrieval," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5552–5561, 2017.
[44] A. K. Bhunia, Y. Yang, T. M. Hospedales, T. Xiang, and Y.-Z. Song, "Sketch less for more: On-the-fly fine-grained sketch-based image retrieval," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9776–9785.
[45] A. K. Bhunia, V. R. Gajjala, S. Koley, R. Kundu, A. Sain, T. Xiang, and Y.-Z. Song, "Doodle it yourself: Class incremental learning by drawing a few sketches," 2022. [Online]. Available: https://arxiv.org/abs/2203.14843
[46] C. Gao, Q. Liu, Q. Xu, L. Wang, J. Liu, and C. Zou, "SketchyCOCO: Image generation from freehand scene sketches." [Online]. Available: http://arxiv.org/abs/2003.02683
[47] C. Zou, Q. Yu, R. Du, H. Mo, Y.-Z. Song, T. Xiang, C. Gao,

B. Chen, and H. Zhang, "SketchyScene: Richly-annotated scene sketches." [Online]. Available: http://arxiv.org/abs/1808.02473
[48] H. Wang, S. Ge, Z. Lipton, and E. P. Xing, "Learning robust global representations by penalizing local predictive power," in *Advances in Neural Information Processing Systems*, 2019, pp. 10 506–10 518.
[49] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 31, no. 4, pp. 44:1–44:10, 2012.
[50] S. Cabi, S. G. Colmenarejo, A. Novikov, K. Konyushkova, S. Reed, R. Jeong, K. Zolna, Y. Aytar, D. Budden, M. Vecerik, O. Sushkov, D. Barker, J. Scholz, M. Denil, N. de Freitas, and Z. Wang, "Scaling data-driven robotics with reward sketching and batch reinforcement learning," *arXiv preprint arXiv:1909.12200*, 2019.
[51] K. Desai, G. Kaul, Z. Aysola, and J. Johnson, "RedCaps: web-curated image-text data created by the people, for the people." [Online]. Available: http://arxiv.org/abs/2111.11431
[52] Y. Lu, S. Wu, Y.-W. Tai, and C.-K. Tang, "Image generation from sketch constraint using contextual GAN." [Online]. Available: http://arxiv.org/abs/1711.08972
[53] Z. Li, C. Deng, E. Yang, and D. Tao, "Staged sketch-to-image synthesis via semi-supervised generative adversarial networks," *IEEE Transactions on Multimedia*, vol. 23, pp. 2694–2705, 2021.
[54] T. Portenier, Q. Hu, A. Szabó, S. A. Bigdeli, P. Favaro, and M. Zwicker, "Faceshop: Deep sketch-based face image editing," 2018. [Online]. Available: https://arxiv.org/abs/1804.08972
[55] K. Olszewski, D. Ceylan, J. Xing, J. Echevarria, Z. Chen, W. Chen, and H. Li, "Intuitive, interactive beard and hair synthesis with generative models," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7444–7454.
[56] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching," in *IJCAI*, 1977, pp. 659–663. [Online]. Available: http://ijcai.org/Proceedings/77-2/Papers/024.pdf
[57] A. Noguchi and T. Harada, "Image generation from small datasets via batch statistics adaptation," 2019. [Online]. Available: https://arxiv.org/abs/1904.01774
[58] S. Mo, M. Cho, and J. Shin, "Freeze the discriminator: a simple baseline for fine-tuning gans," 2020. [Online]. Available: https://arxiv.org/abs/2002.10964
[59] M. Zhao, Y. Cong, and L. Carin, "On leveraging pretrained gans for generation with limited data," 2020. [Online]. Available: https://arxiv.org/abs/2002.11810
[60] Y. Wang, A. Gonzalez-Garcia, D. Berga, L. Herranz, F. S. Khan, and J. van de Weijer, "Minegan: Effective knowledge transfer from gans to target domains with few images," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9329–9338.
[61] Y. Li, R. Zhang, J. C. Lu, and E. Shechtman, "Few-shot image generation with elastic weight consolidation," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 15 885–15 896. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/b6d767d2f8ed5d21a44b0e5886680cb9-Paper.pdf
[62] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *CoRR*, vol. abs/1612.00796, 2016. [Online]. Available: http://arxiv.org/abs/1612.00796
[63] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, "Differentiable augmentation for data-efficient gan training," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 7559–7570. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/55479c55ebd1efd3ff125f1337100388-Paper.pdf
[64] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 12 104–12 114. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/8d30aa96e72440759f74bd2306c1fa3d-Paper.pdf
[65] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2962–2971.
[66] Z. Zhao, Z. Zhang, T. Chen, S. Singh, and H. Zhang, "Image augmentations for GAN training," *CoRR*, vol. abs/2006.02595, 2020. [Online]. Available: https://arxiv.org/abs/2006.02595
[67] S.-Y. Wang, D. Bau, and J.-Y. Zhu, "Sketch your own GAN," number: arXiv:2108.02774. [Online]. Available: http://arxiv.org/abs/2108.02774
[68] M. Li, Z. Lin, R. Mech, E. Yumer, and D. Ramanan, "Photo-sketching: Inferring contour drawings from images," 2019. [Online]. Available: https://arxiv.org/abs/1901.00542
[69] K. Crowson, S. Biderman, D. Kornis, D. Stander, E. Hallahan, L. Castricato, and E. Raff, "Vqgan-clip: Open domain image generation and editing with natural language guidance," 2022. [Online]. Available: https://arxiv.org/abs/2204.08583
[70] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop." [Online]. Available: http://arxiv.org/abs/1506.03365
[71] S. Cheema, S. Gulwani, and J. LaViola, "Quickdraw: Improving drawing experience for geometric diagrams," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 1037–1064. [Online]. Available: https://doi.org/10.1145/2207676.2208550
[72] T. Grigoryev, A. Voynov, and A. Babenko, "When, why, and which pretrained GANs are useful?" in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=4Ycr8oeCoIh
[73] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.
[74] O. Gafni, A. Polyak, O. Ashual, S. Sheynin, D. Parikh, and Y. Taigman, "Make-a-scene: Scene-based text-to-image generation with human priors," number: arXiv:2203.13131. [Online]. Available: http://arxiv.org/abs/2203.13131
[75] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," number: arXiv:2012.09841. [Online]. Available: http://arxiv.org/abs/2012.09841
[76] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric." [Online]. Available: http://arxiv.org/abs/1801.03924
[77] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, "StyleCLIP: Text-driven manipulation of StyleGAN imagery." [Online]. Available: http://arxiv.org/abs/2103.17249
[78] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer International Publishing, pp. 818–833.
[79] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common objects in context." [Online]. Available: http://arxiv.org/abs/1405.0312

Safe and Efficient Multi-agent Reinforcement Learning via Dynamic Shielding

Wenli Xiao[1], Yiwei Lyu[2], John M. Dolan[3]

***Abstract*— Multi-Agent Reinforcement Learning (MARL) discovers policies that maximize reward but do not have safety guarantees during learning and deployment phases. Recent works on safe RL approaches focus on single-agent scenarios, which are not scalable for multi-agent tasks. Furthermore, normal decentralized approaches usually incur high coordination costs when agents interact (i.e., agents may behave conservatively and get stuck in place when close to others). In this paper, we propose an adaptive algorithm to achieve safe and efficient MARL by leveraging shielding to monitor unsafe actions and refactoring the shields frequently to mitigate coordination overhead. A shield is a reactive system running in parallel with the environment to monitor and correct agents' behavior. Our algorithm, dynamic shielding, synthesizes multiple shields in a decentralized fashion, in which each shield monitors a subset of agents. The core idea is that the shields are not fixed during learning; instead, each shield can merge with other shields or split into multiple shields. Shields can merge to monitor agents jointly when multiple agents are gathered in one area; they can also split up to monitor agents in decentralized fashion as agents spread out. Theoretically, our approach ensures safe MARL during the training and execution phases. Our approach reduces coordination costs significantly. We provide experiments to demonstrate that our approach leads to less conservative behaviors than the existing approaches. Our approach also significantly outperforms the existing approaches.**

***Index Terms*— Reinforcement Learning, Formal Methods in Robotics and Automation, Multi-Robot Systems**

I. INTRODUCTION

Multi-Agent Reinforcement Learning (MARL) [1], [2] is a promising approach to obtain learning control policies for multi-agent decision-making tasks such as transportation management [3], [4], motion control [5], [6], and autonomous driving [7]–[9]. However, applying MARL methods in safety-critical autonomous systems (e.g., autonomous driving cars) can cause havoc due to the lack of formal safety guarantees. In addition, traditional MARL approaches with behavior penalties (i.e., giving a negative reward for unsafe actions) cannot ensure safety in practice [10], [11]. Therefore, there is a significant challenge to developing safe MARL systems that are provably trustworthy [2], [10]. Recently, there has been much research in notions of safety [11]–[15]. For example, Linear Temporal Logic (LTL) [16] is a specification language used for formal verification to ensure that an automation system always stays in safety states [17]. A recent work [15] adopts LTL as a safety specification language in single-agent Reinforcement Learning (RL) via synthesizing a shield to monitor the RL agent. The shield is a lightweight system running along with the RL agent, which monitors actions selected by the RL agent and rejects any unsafe actions according to the given safety specification. The shield has provable safety guarantees for the lifetime of the RL process (i.e., the training and deployment phases). Factored shielding [11] adapts the shielded learning method to multi-agent scenarios in a decentralized fashion. Compared with centralized shielding, which uses one shield to monitor the states and actions of all agents, factored shielding synthesizes multiple shields, and each shield monitors a subset of the agents' state space.

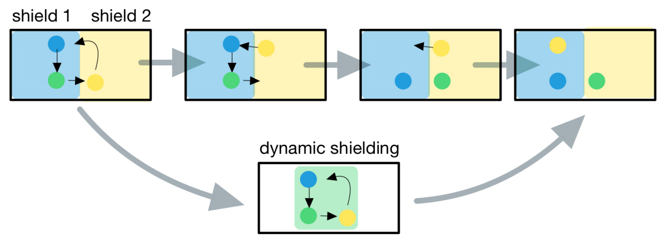


Fig. 1: The differently colored circles denote multiple agents, and the black arrows are desired actions. Traditional decentralized shielding (upper) takes extra steps in waiting for coordination near the border of shields, while proposed dynamic shielding (lower) efficiently takes action.

For multi-agent safe reinforcement learning, there is a dilemma: centralized approaches have limited scalability [11], while fully decentralized methods cause coordination overhead. Agents become stuck waiting for coordination when they get closer to one another due to the lack of information sharing in decentralized approaches. For instance, Figure 1 shows a scenario in which factored shielding causes extra coordination overhead. In this paper, we propose a novel safe and efficient MARL framework in a mixed decentralized manner, which dynamically uses shielding to ensure safety and mitigate conservative behaviors.

Specifically, our main contributions are threefold: Firstly, we propose a novel shield framework - *dynamic shielding*, which enables robots collaboratively to ensure safety. There are initially multiple shields, which concurrently monitor different agents. When there is a high risk of conservative behavior (e.g., agents move together), the shields could choose to merge with others. The merged shield can leverage the

[1]The author is with the School of Science and Engineering, the Chinese University of Hong Kong, Shenzhen. This work was completed when Wenli Xiao served as an intern in the Robotics Institute Summer Scholar at Carnegie Mellon University. wenlixiao@link.cuhk.edu.cn

[2]The author is with the Department of Electrical and Computer Engineering, Carnegie Mellon University. yiweilyu@andrew.cmu.edu

[3]The author is with the Robotics Institute, Carnegie Mellon University. jmd@cs.cmu.edu

state information of multiple agents to mitigate unnecessary coordination overhead. When agents move apart from each other, the merged shield can split into multiple shields to monitor agents. We also present an effective shield synthesis approach in section V, named *k-step look ahead shield*. Our method prunes the unnecessary computation of traditional shield synthesis approaches [11], [15] and delegates the computation complexity to the online algorithm, which can synthesize shields in real-time.

In addition, we showcase the effectiveness and performance of our shielding approach through experiments. We study the navigation problem on six different maps in the grid world [18] and a cooperative environment in the Multiple Particle Environment [19] (MPE). Our approach outperforms all the other benchmarks in terms of reward and minimal steps with a guarantee of safety.

II. Related Works

A. Safe Multi-agent Reinforcement Learning

Safe RL methods can be classified into two categories [12]: 1) The first is optimization criterion-based methods, which modify the RL objective functions. For example, SNO-MDP [20] tackles the safe RL problem using a constrained Markov decision process. 2) The second is based on modifying the exploration process to avoid undesirable actions, which incorporates extra domain-specific knowledge (e.g., guidance and demonstration) into the training process. Our dynamic shielding algorithm falls into the second category. Shielding was introduced to single-agent RL in [15], and was adapted to multi-agent settings in [11]. In this work, we propose a novel shielding framework for MARL by addressing challenges such as coordination overhead and scalability issues in the multi-agent setting.

B. Linear Temporal Logic and Safety Specification

In this paper, we use Linear Temporal Logic (LTL) [16] to write safety specifications. LTL is a widely used specification language in safety-critical systems [21], [22], which can express complex requests at a high level. For example, LTL has been used to express complex task specifications for robotic planning and control [23], [24]. Several works [25]–[27] develop reward shaping techniques that translate logical constraints expressed in LTL to reward functions for RL. However, [11] has empirically demonstrated reward shaping cannot ensure safety in MARL. The shield synthesis technique based on solving two-player safety games was developed in [28] for enforcing safety specifications written by LTL, which synthesizes the shield to a local file before running the system (offline). We modify the two-player game and propose an online method to synthesize shields in real-time (in section V).

III. Preliminaries

We start by introducing *Multi-agent Reinforcement Learning*, *Shielding*, and *Safety Games with Linear Temporal Logic specification*, upon which our algorithm builds.

A. Multi-Agent Reinforcement Learning (MARL)

We focus on the n-player Markov Games defined by a tuple $\left(\mathcal{N}, \mathcal{S},\left\{\mathcal{A}^{i}\right\}_{i \in \mathcal{N}},\left\{r^{i}\right\}_{i \in \mathcal{N}}, \mathcal{P}, \gamma\right)$, where $\mathcal{N}=\{1 \ldots n\}$ is the set of n agents, $\mathcal{S}$ denotes the state space jointly observed by all agents, $\mathcal{A}^i$ is the action space of agent i, r^i is the reward function of agent i, $\mathcal{P}: \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ denotes the transition probability, and γ is the discount factor. We assume the initial state s_1 follows a fixed distribution $\rho \in \Delta(\mathcal{S})$. At each time step t, the agents observe state s_t, take actions $a_{t,i} \in A^i$ in the environment simultaneously, and receive rewards $r_{t,i} \in R^i$. Then the state of the environment moves to s_{t+1}. The objective of each agent i is to learn a control policy π_i which maximizes the expected cumulative reward $E\left[\sum_{t=0}^{\infty} \gamma^{t} R^{i}\left(s_{t}, a_{t}, s_{t+1}\right)\right]$. MARL algorithms can be categorized into three different types based on the dependence of individual agent performance on other agents' choices, including cooperative, competitive, and mixed settings. We use MARL algorithms with mixed settings in our experiment, and discuss the details in Section VI.

B. Linear Temporal Logic as Safety Specification

We consider Linear Temporal Logic [16] (LTL) to express safety specifications. LTL is an extension of propositional logic, which has long been used as a tool in the formal verification of programs and systems. The syntax of LTL is given by the following grammar [29]:

$$\varphi := p|\neg p|\varphi_1 \vee \varphi_2| \bigcirc \varphi|\varphi_1 \mathcal{U} \varphi_2$$

where p is an atomic proposition. The temporal operators are next $\bigcirc\varphi$, which indicates φ is true in the next succeeding state, and until $\varphi_1 \mathcal{U} \varphi_2$ indicating φ_1 is true until the state where φ_2 is true. From these operators, we can define $True \equiv \phi \vee \neg\phi$, $False \equiv \neg True$, implication $\varphi \Rightarrow \psi := \neg\varphi \vee \psi$, eventually $\diamond\varphi := \mathrm{True}\,\mathcal{U}\varphi$, and always $\square\varphi := \neg \diamond \neg\varphi$. We use LTL formulas to express safe specifications. For example, $\square\neg collision$ denotes that collision should never happen. We consider translating the LTL safety specification into a safe language accepted by a deterministic finite automaton (DFA) [30]. In addition, we extend the definition of safe RL in [15] to MARL in the following way:

Definition 1. *Safe MARL is the process of learning optimal policies for multiple agents while satisfying a temporal logic safety specification ϕ^s during the learning and execution phases.*

C. Formal Safety Guarantee with Shield

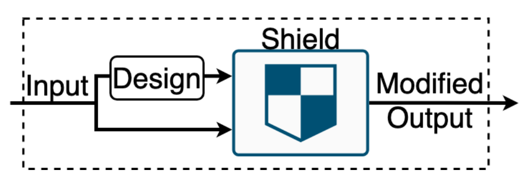


Fig. 2: Enforcing safety specification via shield.

Our method builds upon a prior method called Shield [11], [28], which ensures safety properties at runtime. Shield (Fig. 2) monitors the control input of agents and corrects any unsafe control input instantaneously. A Shield should have two properties: 1) Minimal interference. Namely, shields only correct the action if it violates the safety rule. 2) Correctness. Shields should distinguish every unsafe action and refine it with safe actions. Our method uses the Shield framework to ensure safety, and we provide theoretical proof of safety in section IV.

We **represent the shield** using a finite-state reactive system. According to the formulation in [11], a finite-state reactive system is a tuple $S = (Q, q_0, \Sigma_I, \Sigma_O, \delta, \lambda)$, where Σ_I and Σ_O are the I/O alphabets, Q is the state set, $q_0 \in Q$ denotes the initial state, $\delta : Q \times \Sigma_I \rightarrow Q$ is a transition function, and $\lambda : Q \times \Sigma_I \rightarrow \Sigma_O$ is an output function. Given the symbolic abstraction of the control input (i.e., input trace) $\overline{\sigma_I} = x_0 x_1 \ldots \in \Sigma_I^{\infty}$, the system S generates the trajectory of states (i.e., output trace) $\overline{\sigma_O} = \mathcal{S}\left(\overline{\sigma_I}\right) = \lambda\left(q_0, x_0\right) \lambda\left(q_1, x_1\right) \ldots \in \Sigma_O^{\infty}$, where $q_{i+1} = \delta(q_i, x_i)$ for all $i \geq 0$.

We **synthesize the shield** by solving a *two-player safety game* [28], a game played by the MARL agents and the environment, where the winning condition is defined by the LTL safety specification. MARL agents should comply with all safety specifications all of the time in order to win the game. A two-layer game is a tuple $\mathcal{G} = (G, g_0, \Sigma_I, \Sigma_O, \delta, win)$ with a finite set of game states G, the initial state $g_0 \in G$, a complete transition function $\delta : G \times \Sigma_I \times \Sigma_O \rightarrow G$, and win as a winning condition. In every state $g \in G$, the environment first chooses an input action $\sigma_I \in \Sigma_I$, and then the MARL agents choose a joint action (in abstraction symbol) $\sigma_O \in \Sigma_O$. Then the game moves to the next state $g' = \delta(g, \sigma_I, \sigma_O)$, and so forth. The resulting trajectory of game states $\overline{g} = g_0, g_1, \ldots$ is called a *play*. A play is won if and only if $win(\overline{g})$ is $true$. We describe the detailed procedure of synthesizing shields via solving the two-player safety game in section V.

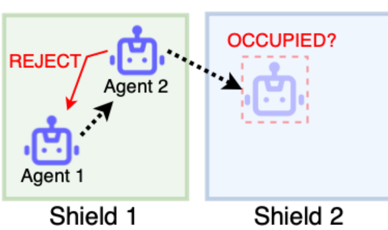


Fig. 3: The large squares denotes shields, and the dashed arrows are desired actions of agents. Shield 1 conservatively judges agent 2 cannot successfully enter shield 2, thus rejecting Agent 1's action.

IV. Tackling Safe and Efficient Multi-agent Reinforcement Learning via Dynamic Shielding

In this section, we first describe how traditional shielding methods cause learning to be inefficient. Then, we present our method for safe and efficient MARL learning.

A. Conservative Behavior and Coordination Overhead

For multi-agent systems, centralized approaches always fail when the number of agents increases. For example, centralized shielding for MARL fails empirically for two-agent scenarios [11]. Fully decentralized shielding separates the whole state space into exclusive subspaces and synthesizes a shield to monitor a subspace. For example, factored shielding [11] computes multiple shields based on a factorization of the joint state space observed by all agents. However, this approach causes conservative behavior (i.e., agents stuck in place) when agents move across the border of shields due to the information isolation between shields. Specifically, as shown in Figure 3, the shield would reject agents' actions even for those are essentially valid. As a consequence, the MARL system needs higher coordination overhead, say extra steps, when agents have interaction. In section VI, we empirically demonstrate that the coordination overhead caused by conservative behaviors leads to sub-optimal policies for MARL agents.

B. Dynamic shielding

Algorithm 1: Dynamic Shielding

Input: A list of shields $S = \{s_1, s_2, ..., s_m\}$, MARL agents' joint action $a_t = (a_t^1, a_t^2, ...a_t^n)$ and joint state $s_t = (s_t^1, s_t^2, ..., s_t^n)$, a constant penalty for unsafe actions p
Output: Safe joint action $\bar{a}_t$, punishment p_t, shield new_shield

/* Clustering: divide agents into groups by some methods */
new_shield $=$ *cluster_agents*(s_t, a_t)
for *all group in* $i \in \{1, ..., m'\}$ **do**
    for *all shield* $j \in \{, ..., m\}$ **do**
        if *new_shield[I].group* == s_j*.group* *and* s_j*.duration* $!=$ *0* **then**
            new_shield[i].recompute = False
            new_shield[i].shield = $s_j.shield$
        end if
    end for
end for
/* Re-construct shields */
for *all group in* $i \in \{1, ..., m\}$ **do**
    if $new\_shield[i].recompute == True$ **then**
        new_shield[i].reCompute()
    end if
end for
/* Shielding */
$\bar{a}_t$ = safe action output by new_shield
for *all agent in* $i \in \{1, ..., n\}$ **do**
    if $\bar{a}_t^i \neq a_t^i$ **then**
        p_t^i = p
    end if
end for
return *bar*a_t, p_t, *new_shield*

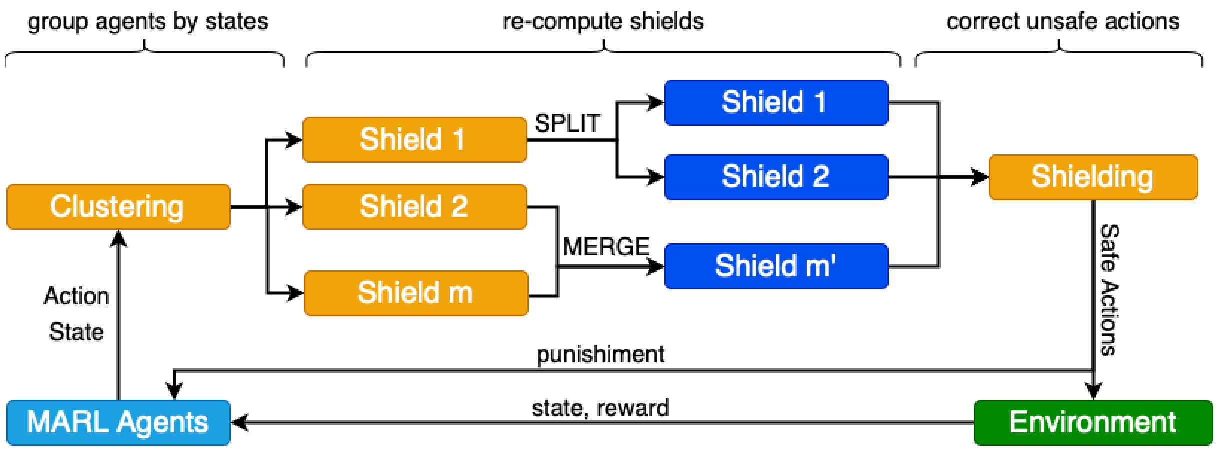


Fig. 4: Safe MARL with dynamic shielding

To mitigate the coordination overhead caused by conservative behaviors, we propose *dynamic shielding*, a decentralized shield framework for the traditional MARL process. Dynamic shielding has two important features: 1) Dynamic shielding dynamically constructs shields based on agents' real-time states; 2) Dynamic shielding can perform two extra operations, namely, *merge* and *split*. The merge operation uses multiple shields' information to construct a larger shield, which temporarily removes the border between shields. Therefore, the merged shield has enough information to distinguish whether actions are safe, and eventually mitigate the conservative behavior. On the other hand, the computation complexity in shield synthesis increases along with the shield size. The split operation helps decrease computation costs when agents locate sparsely. Figure 4 shows the diagram of dynamic shield construction. Initially, we construct distinct shields for each agent, which monitor agents' reachable states in the next k steps. If agents try to move to states outside the shield, the shield will recompute to establish a monitor on agents' future possible states. When agents gathering together has the possibility of collision, shields will merge to jointly monitor the action using locomotion and dynamics of multiple agents. When agents are more sparse, the merged shield will split to save computation.

We summarize dynamic shielding in algorithm 1. There are three phases: 1) clustering, 2) shield reconstruction, and 3) shielding. In the clustering phase (LINE 2-11), the algorithm clusters agents into groups by their current state. For example, in robot navigation tasks, if some agents are close by, the algorithm will put them in the same group, otherwise in separate groups. Agents in the same group should merge shields to avoid conservative behaviors. Then, in the shield re-construction phase (LINE 13-17), shields will merge with other shields or split into multiple smaller shields based on the results of clustering. In addition, some expired shields might recompute according to agents' state change. In the shielding phase (Lines 19-23), every shield will do shielding concurrently, which rejects agents' unsafe actions and replaces them with safety actions. Lastly, the MARL agents will be given an extra penalty for unsafe actions.

Our method faces the challenge that it degrades to centralized shielding for edge scenarios. For example, when all agents gather together, all decentralized shields will merge together into a single centralized shield. We propose a new way of shield synthesis in Section V and provide experiments in Section VI to demonstrate that even in the worst case, our method is still more scalable than centralized shielding.

V. Synthesize Shield in Real-Time

In this section, we present our shield synthesis method – *k-step look ahead shield*, a variant of traditional shield synthesis [28]. We also give theoretical proof to show that our method guarantees safety.

A. *k-step look ahead shield*

We assume the state space has been converted into a symbolic abstraction given by a DFA $\mathcal{A}^e = (Q^e, q_0^e, \Sigma^e, \delta^e, F^e)$. We translate the LTL safety specification into another DFA $\mathcal{A}^S = (Q^S, q_0^S, \Sigma^S, \delta^S, F^S)$. We formulate a two-player game $\mathcal{G} = (G, g_0, \Sigma_1, \Sigma_2, \delta^g, F)$ by combining $\mathcal{A}^e$ and $\mathcal{A}^S$. Instead of solving the game G directly, we add extra time constraints $t \leq k$, where $t \in$ denotes the time step from constructing the shield, and k is a hyper-parameter that denotes the maximum steps of the game. The modified game is then

$$\mathcal{G}^k = \left(G^k, {g_0}', \Sigma_1, \Sigma_2, {\delta^g}', F^k\right) \tag{1}$$

where the state space $G^k = G \times \{1...k\}$, the initial state ${g_0}' = (g_0, t = 1)$, the transition function ${\delta^g}'(g_t, t) = (\delta^g(g_t), t+1)$, and the winning condition $F^k = F \wedge (t \leq k)$. We can solve the two-player safety game $\mathcal{G}^k$ and compute the winning region $W \subseteq F^k$, using the method in [28]. We then construct the *k-step look ahead shield* by translating $\mathcal{G}^k$ and W to a reactive system $S = (Q_\mathcal{S}, q_{0,\mathcal{S}}, \Sigma_{I,\mathcal{S}}, \Sigma_{O,\mathcal{S}}, \delta_\mathcal{S}, \lambda_\mathcal{S})$. The shield has the following components: $Q_\mathcal{S} = G^k$, $q_{0,\mathcal{S}} = {q_0}'$, $\Sigma_{I,\mathcal{S}} = L \times \mathcal{A}$, $\Sigma_{O,\mathcal{S}} = \mathcal{A}$, $\delta_\mathcal{S}(g^k, (l, a)) =$

$\delta\left(g^k,(l,\lambda_{\mathcal{S}}(g,(l,a)))\right)$ for all $g^k \in G, l \in L, a \in \mathcal{A}$, and

$$\lambda_{\mathcal{S}}(g,l,a)=\begin{cases} a & \text{if } \delta^k(g^k,(l,a)) \in W \\ a' & \text{if } \delta^k(g^k,(l,a)) \notin W \text{ for some arbitrary} \\ & \text{default } a' \text{ with } \delta^k\left(g^k,(l,a')\right) \in W. \end{cases}$$

Our shield synthesis bears a resemblance to the classic shield synthesis [11], [15], which also synthesizes shields by solving the two-player game. The main difference is our method only predicts a subset of future state space, whereas previous methods enumerate all possible states along the planning horizon. This leads to the major benefit of our method, that for tasks that state spaces too large to compute in advance, our algorithm still works efficiently while previous methods fail.

B. Safety Guarantee

We show that dynamic shielding with *k-step look ahead shielding* can guarantee safety for MARL agents.

Proposition 1. *Given a trace $s_0a_0s_1a_1\cdots \in (S \times A)^\omega$ jointly produced by MARL agents, the dynamic shielding, and the environment, state-action pair (s_t,a_t) is safe at every time step regarding definition 1.*

Proof: Firstly, the procedure in algorithm 1 ensures each agent is monitored by a shield at each time step, and this shield at least monitors the states of agents in the next k steps (otherwise, the shield will re-compute). Then the remaining proof is the same as the correctness of centralized shielding in [11]. For any agents under shield $\mathcal{S} = (Q, q_0, \Sigma_I, \Sigma_O, \delta, \lambda)$, there is a corresponding run $q_0q_1,...q_m \in (S \times A)^\omega$, where $m \leq k$ is the duration before reconstructing this shield. By constructing the shield, we have the environment abstraction DFA A^e and the safety specification DFA A^s. We can project the run $q_0,q_1,...q_m$ of the shield $\mathcal{S}$ onto a trace $q_0^s(f(s_0),a_0)q_1^s(f(s_1),a_1)...q_m^s(f(s_m),a_m)$ on A_s. Since we construct the shield from the winning region of the two-player safety game, every state $q_i^s(f(s_i)$ visited by agents along the trace should be a safe state in $\mathcal{A}^s$. The shield $\mathcal{S}$ ensures the safety specification defined in $\mathcal{A}^s$ is never violated. Therefore, the joint state-action pair (s_t,a_t) is safe for every MARL agent at every step.

VI. Experiments

In this section, we empirically evaluate the performance of our proposed safe MARL framework (Algorithm 1) and compare it with other benchmarks. We apply our algorithm to six benchmark problems in the gridworld (Fig 5) and a cooperative environment of MPE (Fig 6). We compare the proposed algorithm with CQ-Learning [31], CQL with factored shielding, DDPG [32], and MADDPG [19]. We implement algorithms using Python and synthesize shields via Slugs [33]. For each experiment, we evaluate algorithms in both training and testing phases. To mitigate the outliers, we conduct all experiments in 20 independent runs and average the results.

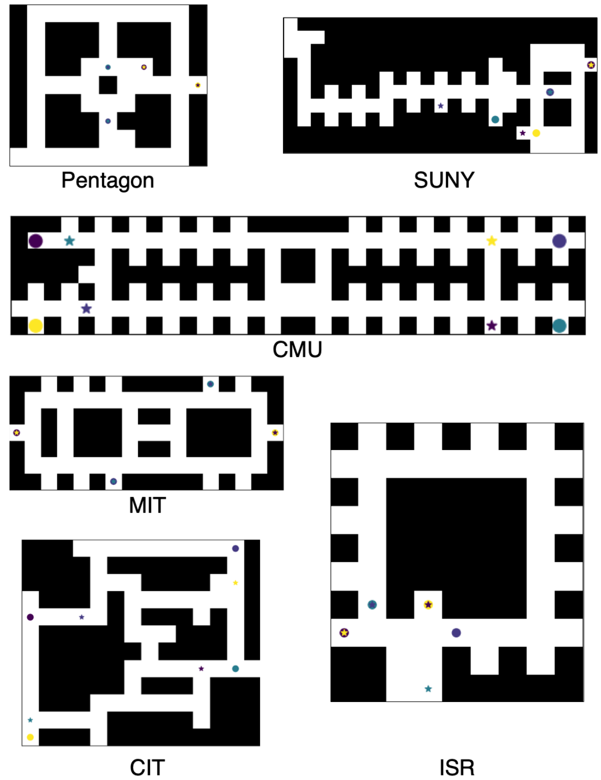


Fig. 5: Different gridworld environments. Dots are agents, stars denote targets, and black blocks are obstracles.

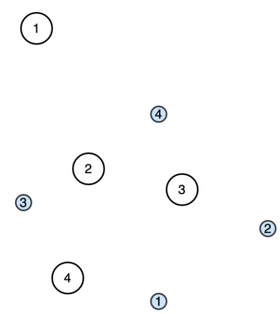


Fig. 6: Adapted *simple spread* MPE environment. The environment is unbounded but agents will be given a penalty when moving far from the center of the map.

Experiment Setup. Figure 5 shows six maps of grid world benchmark environments adapted from [34]. Each map has two agents learning to navigate while avoiding obstacles in the environment. Each agent has the action space $\mathcal{A} = \{stay, up, down, left, right\}$. We assign a unique target to each agent. Once an agent reaches its target, it stays there until all agents reach their goals. We set sparse goal-reaching rewards for this task, namely, giving -1 living penalty, -10 collision penalty, and $+100$ for arriving the target.

Figure 6 shows a cooperative navigation task adapted from MPE [19] *simple spread* environment. The goal is for agents to cooperate and reach their target while avoiding collisions. This task is more difficult than the gridworld in two aspects:

1) The state space is continuous and unbounded.

Map \Algorithm	**CQL + Dynamic shielding (Proposed)**		CQL + Factored shielding		CQL w/o Shielding	
	Reward	Collision	Reward	Collision	Reward	Collision
Pentagon	**88**	**0**	85	0	88	266
CIT	**80**	**0**	73	0	79	447
MIT	**79**	**0**	74	0	61	180
CMU	**64**	**0**	39	0	47	184

TABLE I: Results compare the CQ-learning, CQ-learning with factored shielding, and CQ-learning with dynamic shielding.

2) Agents have more complicated dynamics, such as momentum and acceleration.

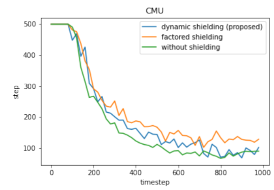
Each agent has the action space

$$\mathcal{A} = \{stay, up, down, left, right, brake\},$$

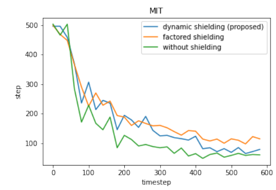
from which the action controls acceleration. For example, when an agent performs $stay$, it moves at its original velocity. We use $brake$ to simulate braking in the real world, where the agent exerts a large deceleration in the opposite direction of velocity until it fully stops. The $brake$ action obeys the law of kinematics; for example, an agent moving at a higher speed needs a longer distance to brake down. Each agent receives a reward that is inversely proportional to the distance with its target and a -1 penalty for any collision.

Conservative Behavior Evaluation. We integrate CQ-learning with factored shielding and proposed dynamic shielding. At this stage, we apply them to four of the grid-world environments (Fig 5). We evaluate algorithms using the metrics such as maximum rewards, collision counts, and episode steps during the training phase. Results in table I show that factored shielding and dynamic shielding can guarantee collision-free learning in all maps. Moreover, dynamic shielding obtains better policies with higher rewards compared to factored shielding and no shielding. Figure 8 shows agents using proposed dynamic shielding need fewer steps to reach the target than factored shielding. Besides, dynamic shielding policy eventually has comparable performance as CQ-learning without intervention, which we consider as the ground truth regarding the steps to reach the target. Therefore, the proposed dynamic shielding mitigates the conservative behaviors while ensuring safety.

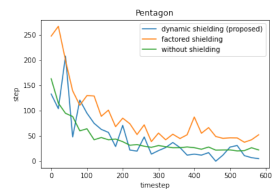
Scalability Evaluation. At this stage, we evaluate the performance of the proposed dynamic shielding when the state space scales up. We integrate DDPG and MADDPG with dynamic shielding and apply them to the adapted MPE environment shown in Fig 6. Factored shielding fails in this unbounded environment since we cannot synthesize shields for the entire state space. Figure 8b shows dynamic shielding can guarantee no collision during the training process. Whereas DDPG and MADDPG constantly have collisions even at the end of training. Figure 8a shows the episode rewards per training step (i.e. episode). At convergence, MADDPG with dynamic shielding and DDPG with dynamic shielding have the highest reward than vanilla MADDP and DDPG without shielding. The learning curves in Figure 8a demonstrate that proposed dynamic shielding improves the performance of MARL. In the future, we will evaluate the performance of the proposed algorithm when the number of agents scales up.



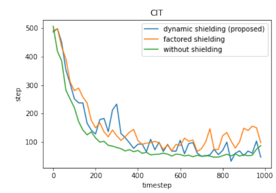
(a) CMU



(b) MIT

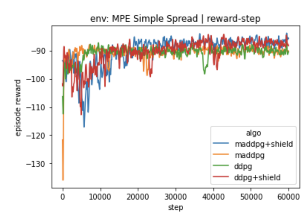


(c) Pentagon

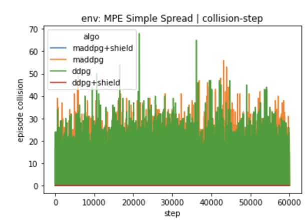


(d) CIT

Fig. 7: Experiment results on gridworld.



(a) Episode reward



(b) Episode collision

Fig. 8: Experiment results on MPE.

VII. CONCLUSIONS

This paper presents a novel approach to safe MARL via dynamic shielding. Our proposed method minimally interferes with the MARL framework to ensure the safety specification defined by LTL expressions. We also propose an effective technique to synthesize shields in real-time and provide theoretical proof of a safety guarantee. Besides, we conduct extensive experiments to demonstrate our algorithm is better than other shielding approaches regarding efficiency and improves learning performance in complex environments.

ACKNOWLEDGMENT

This work is supported by the Robotics Institute Summer Scholars (RISS) program at Carnegie Mellon University, the Chinese University of Hong Kong, Shenzhen (CUHK, Shenzhen), and the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS). The authors would like to thank Rachel Burcin and Prof. John Dolan for their support and organization of the RISS program.

References

[1] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.

[2] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of Reinforcement Learning and Control*, pp. 321–384, 2021.

[3] L. Okdinawati, T. M. Simatupang, and Y. Sunitiyoso, "Multi-agent reinforcement learning for value co-creation of collaborative transportation management (ctm)," *International Journal of Information Systems and Supply Chain Management (IJISSCM)*, vol. 10, no. 3, pp. 84–95, 2017.

[4] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 128–135, 2010.

[5] A. Perrusquía, W. Yu, and X. Li, "Redundant robot control using multi agent reinforcement learning," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2020, pp. 1650–1655.

[6] C. Yu, X. Wang, and Z. Feng, "Coordinated multiagent reinforcement learning for teams of mobile sensing robots," in *Proceedings of the 18th international conference on autonomous agents and multiagent systems*, 2019, pp. 2297–2299.

[7] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, 2016.

[8] S. Bhalla, S. Ganapathi Subramanian, and M. Crowley, "Deep multi agent reinforcement learning for autonomous driving," in *Canadian Conference on Artificial Intelligence*. Springer, 2020, pp. 67–78.

[9] M. Zhou, J. Luo, J. Villella, Y. Yang, D. Rusu, J. Miao, W. Zhang, M. Alban, I. Fadakar, Z. Chen, *et al.*, "Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving," *arXiv preprint arXiv:2010.09776*, 2020.

[10] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan, "Learning safe multi-agent control with decentralized neural barrier certificates," *arXiv preprint arXiv:2101.05436*, 2021.

[11] I. ElSayed-Aly, S. Bharadwaj, C. Amato, R. Ehlers, U. Topcu, and L. Feng, "Safe multi-agent reinforcement learning via shielding," *arXiv preprint arXiv:2101.11196*, 2021.

[12] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.

[13] M. Pecka and T. Svoboda, "Safe exploration techniques for reinforcement learning–an overview," in *International Workshop on Modelling and Simulation for Autonomous Systems*. Springer, 2014, pp. 357–375.

[14] M. Hasanbeig, A. Abate, and D. Kroening, "Cautious reinforcement learning with logical constraints," *arXiv preprint arXiv:2002.12156*, 2020.

[15] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[16] A. Pnueli, "The temporal logic of programs," in *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*. ieee, 1977, pp. 46–57.

[17] K. Y. Rozier, "Linear temporal logic symbolic model checking," *Computer Science Review*, vol. 5, no. 2, pp. 163–203, 2011.

[18] F. S. Melo and M. Veloso, "Learning of coordination: Exploiting sparse interactions in multiagent systems," in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 2009, pp. 773–780.

[19] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Neural Information Processing Systems (NIPS)*, 2017.

[20] A. Wachi and Y. Sui, "Safe reinforcement learning in constrained markov decision processes," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9797–9806.

[21] R. Alur, *Principles of cyber-physical systems*. MIT press, 2015.

[22] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.

[23] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE transactions on robotics*, vol. 25, no. 6, pp. 1370–1381, 2009.

[24] A. Ulusoy, S. L. Smith, X. C. Ding, C. Belta, and D. Rus, "Optimality and robustness in multi-robot path planning with temporal logic constraints," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 889–911, 2013.

[25] A. K. Bozkurt, Y. Wang, M. M. Zavlanos, and M. Pajic, "Control synthesis from linear temporal logic specifications using model-free reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 349–10 355.

[26] E. M. Hahn, M. Perez, S. Schewe, F. Somenzi, A. Trivedi, and D. Wojtczak, "Omega-regular objectives in model-free reinforcement learning," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2019, pp. 395–412.

[27] M. Hasanbeig, A. Abate, and D. Kroening, "Cautious reinforcement learning with logical constraints," *arXiv preprint arXiv:2002.12156*, 2020.

[28] B. Könighofer, M. Alshiekh, R. Bloem, L. Humphrey, R. Könighofer, U. Topcu, and C. Wang, "Shield synthesis," *Formal Methods in System Design*, vol. 51, no. 2, pp. 332–361, 2017.

[29] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.

[30] O. Kupferman and M. Y. Vardi, "Model checking of safety properties," *Formal methods in system design*, vol. 19, no. 3, pp. 291–314, 2001.

[31] Y.-M. De Hauwere, P. Vrancx, and A. Nowé, "Learning multi-agent state space representations," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, 2010, pp. 715–722.

[32] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[33] R. Ehlers and V. Raman, "Slugs: Extensible gr (1) synthesis," in *International Conference on Computer Aided Verification*. Springer, 2016, pp. 333–339.

[34] F. S. Melo and M. Veloso, "Learning of coordination: Exploiting sparse interactions in multiagent systems," in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. Citeseer, 2009, pp. 773–780.

[35] W. Schwarting, J. Alonso-Mora, L. Pauli, S. Karaman, and D. Rus, "Parallel autonomy in automated vehicles: Safe motion generation with minimal intervention," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1928–1935.

Container Invariant Classification of Substrates Using Spectroscopy

Wesley Lewis[1], Kavya Puthuveetil[2], Akhil Padmanabha[2], Zackory Erickson[2]

***Abstract*— Material classification can provide robots with data that can better inform how they should interact with objects in their environment. Particularly in human-robot interaction, this information may be vital for successful task completion. Material recognition becomes a challenge with current approaches such as visual or haptic sensing when a material is within a container that is opaque or when two materials are visually indistinguishable from one another. This paper shows how Near-Infrared (NIR) Spectroscopy enables the estimation of substrates encased in containers. We collect a dataset of spectral measurements from 36 household substrates (foods and medicine) in containers made from materials such as silicon, plastics, and glass using a miniaturized NIR Spectrometer. We train and compare three different classifiers on this data using Stratified K-fold and Leave-One-Group-Out cross validation.**

I. INTRODUCTION

Robots need to understand the objects in their environment to make informed decisions that prevent errors in manipulation of those object or materials, particularly when handling liquids and/or working around people. Many previous works have presented methods for materials classification including tactile sensing, computer vision, and spectroscopy [1].

In a household, people frequently deal with materials that are inside of containers. These containers may be opaque or the materials inside these containers may be impossible to identify visually, making it challenging for robots to identify materials in their environment using vision alone. In this work, we explore how NIR Spectroscopy can enable material classification in these scenarios where vision or tactile sensing may fail.

Spectroscopy is the measurement of electromagnetic radiation in matter. A spectrometer, the primary tool used in spectroscopy, measures the amount of light reflected from a substrate. The light waves create a frequency spectrum and are used to identify a material. This paper will utilize a subsection of spectroscopy called Near-infrared spectroscopy (NIRS), which only measures light from the near-infrared region of the electromagnetic spectrum.

Using a is the Mantispectra Spectrapod, shown in Fig. 1. We collect a dataset of spectral measurements of various substrates in containers of different colors, materials, opacity, and geometry. Once the data was collected, we trained a support vector machine (SVM), Multilayer Perceptron (MLP), and Random Forest and validated our models using Stratified K-fold and Leave-One-Group-Out Cross Validation.

Fig. 1. Mantispectra Spectrapod in enclosure

In this work we make the following contributions:

- We introduce an approach for substrate classification that is invariant to container type
- We provide a dataset of spectral measurement from a variety of household substrates (liquid/solid) in various containers
- We train three different classifiers on the collected dataset and evaluate their performance via Stratified K-fold Cross Validation for substrate invariant classification and Leave-One-Group-Out Cross Validation for generalization

II. RELATED WORK

A. Spectroscopy

NIR spectroscopy is a spectroscopic technique that uses electromagnetic radiation in the 700 to 2,500 nm range (between visible red light and the mid-infrared region) to analyze structural and compositional characteristics of samples [2]. This technique has been well demonstrated for commercial use in agriculture, pharmaceuticals, food quality assurance, and recyclable classification, favored for its non-destructive nature and low sample preparation. [2], [3]. Prior research has shown how robots can leverage near-infrared spectroscopy to classify household objects [1], [4]. In this work, we use a miniaturized NIR spectrometer (Mantispectra

[1]Wesley Lewis studies computer science at the University Of Virginia, Charlottesville, USA wjl4jj@virginia.edu
[2]Kavya Puthuveetil, [2]Akhil Padmanabha, [2]Zackory Erickson are with the Robotic Caregiving and Human Interaction Lab, Carnegie Mellon University, Pittsburgh, PA, USA {zerickson, kavyap, akhilpad}@andrew.cmu.edu

< TABLE OF CONTENTS

Spectrapod) which has a sensing range of 850 to 1700 nm and is notable for its cost effectiveness and potential to be integrated into a robot's end effector [5], [6].

B. Material Recognition

Material recognition techniques generally fall into two different categories: tactile sensing and computer vision. For tactile sensing systems, haptic sensors provide force, temperature, temperature, and vibrational data that can be used to classify materials. Gelsight sensors [7] have even been demonstrated in viscosity and volume estimation for classification of liquids [8]. Tactile sensing approaches require direct contact with object being classified. For visual systems, the most common system of classification is computer vision.

Prior work has shown how spectroscopy is promising for household material classification [1] and how NIR spectroscopy can be combined with high-resolution texture imaging for a multi-modal classification approach that leverages the strengths of both techniques [4]. In this work, we pursue NIR spectroscopy for substrate classification since spectral signals can provide unique compositional information for materials that are otherwise visually indistinguishable. Additional NIR is capable of penetrating through containers, allowing for substrate sensing in more natural human environments where containers may vary in opacity

Solids	Liquids
Cashews	Red Gatorade
Almonds	Blue Gatorade
Coffee	Almond milk
Lucky Charms	Cows Milk
Cocoa Puffs	Water
Cheerios	Salt Water
Salt	Sugar Water
Sugar	Olive Oil
Baking soda	Vegetable Oil
Ibuprofen	Sesame Oil
Benadryl	Coke
Aleve	Sprite
Tylenol	Isopropyl Alcohol
Melatonin	Povidone Iodine
Oatmeal	Bleach
Rice	Tide
	Chicken Broth
	Beef Stock
	Vinegar
	Soy Sauce

Fig. 2. List of substrates. First column is solids and the second column is liquids

III. DATASET

In this section, we describe our methods and materials used for dataset collection, including the spectrometer used and selected substrates and containers. We also describe our approach for standarizing the data before use in classifier training.

A. Spectrometer

In this paper we use a handheld NIR spectrometer called the Spectrapod from Mantispectra to capture spectral measurements Fig. 1. The spectrometer measures light spectra from wavelengths of 850-1700 nm. Spectral measurements are represented as a 16 bit array where each pixel has a different wavelength response [5]. The Mantispectra Spectrapod comes in an enclosure with optics and illumination embedded in the case.

B. Data Collection

Before collecting data from a full array of substrates and containers, we first evaluated which container materials produce unique spectral signals, indicating that they may be suitable for sensing substrates inside the container. Our initial experiments found that metals produce identical, constant spectral signals regardless of the type of metal. This showed that, broadly, metals are too reflective and that NIR is unlikely to be able to penetrate through metal containers. Based on these findings, we exclude all metal containers from our substrate-container data collection phase.

In our selection of various substrates, we considered various commnon solid and liquid household foods, cleaning supplies, and over-the-counter medication. Some of the selected substrates are indistinguishable visually, for example, cow's milk and almond milk (Fig. 3). We also select a variety of containers that have different color, geometry, opacity, and thickness. In total, our dataset consists of spectral measurements from 16 containers, 20 liquid substrates, and 16 solid substrates. Fig. 2 shows the complete list of substrates and Fig. 4 shows a complete list of containers and their characteristics.

After taking spectral measurements of the container while empty, we then take measurements of each substrate inside each container. The cuvette is an exception and only contains liquid substrates due to its size. During the data collection process, we hold the container within 1 cm of the spectrometer take ten consecutive measurements for each sample. From all the samples combined, we take 5760 measurements.

C. Standardization

Once the data is collected, the data is standardized using the following equation:

$$S_{calspec} = \frac{S_{spec} - \bar{D}_{spec}}{\bar{L}_{spec} - \bar{D}_{spec}} (1)$$

$\bar{L}_{spec}$ refers to the measurement taken from the white block provided by Mantispectra, as seen in Fig. 5, which represents maximum reflectivity, $\bar{D}_{spec}$ refers to the measurement taken

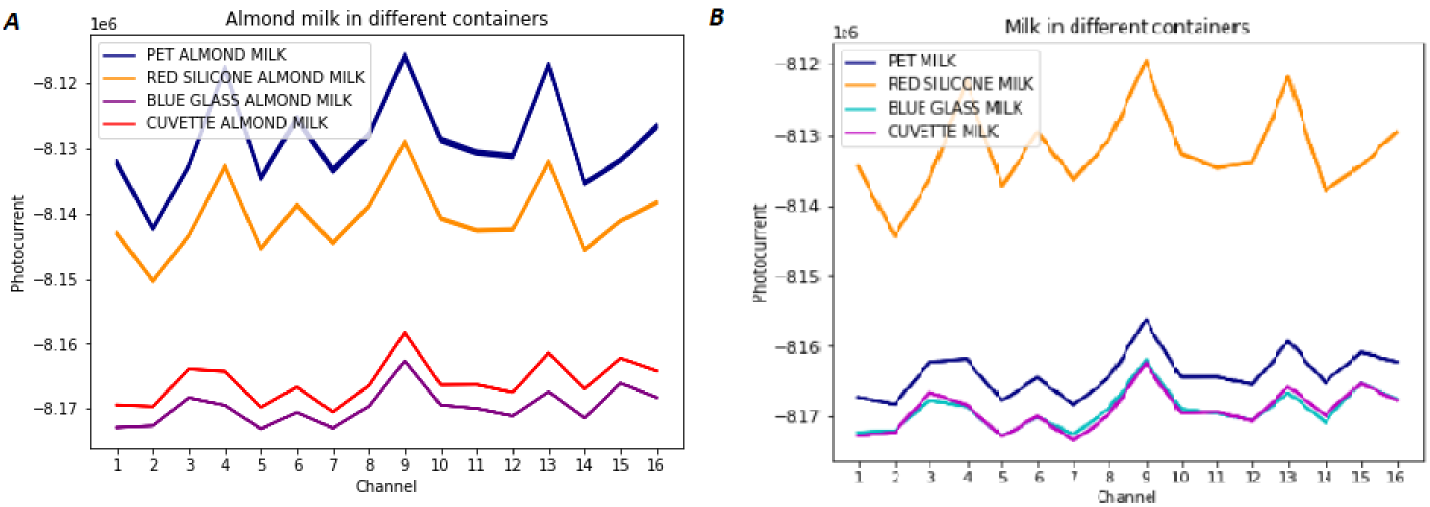


Fig. 3. A: a photocurrent by channel plot of almond milk between four containers, pet, red silicone, blue glass, and cuvette. B: a photocurrent by channel plot of milk between four containers, pet, red silicone, blue glass, and cuvette.

Material	Geometry	Opacity
Silicone	Planar	Opaque
PP (Pill Bottle)	Cylindrical	Transparent
PP (Storage)	Planar	Translucent
HDPE	Cylindrical	Translucent
Acrylic	Planar	Transparent
Styrofoam	Planar	Opaque
PET	Cylindrical	Transparent
Glass (Clear)	Cylindrical	Transparent
Glass (Blue)	Cylindrical	Translucent
Quartz	Planar	Transparent

Fig. 4. Containers and opacity

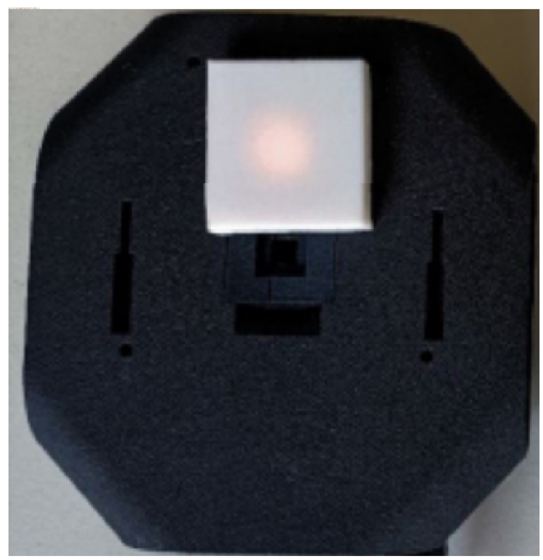


Fig. 5. Mantispectra Spectrapod with LED on taking a measurement of the included white block for white standardization

in a dark room with nothing directly in front of the spectrometer. S_{spec} refers to a spectral signal for a particular sample and S_{cal_spec} refers to the standardized result.

IV. EVALUATION

We train and compare three different models, SVM, Random Forest, and MLP, on the collected dataset to classify the substrate given a measurement of a substrate in some container. We evaluate these classifiers via two cross validation approaches, Stratified K-fold and Leave One Group Out. In Stratified K-Fold Cross Validation,we assess container invariant classification accuracy. In Leave One Group Out Cross Validation we to assess generalization.

A. SVM

We implement a simple SVM using Scikit-learn. The classifier has a Radial basis function kernel. The results for Stratified K-fold using SVM shows the model can estimate containers it has seen before with an accuracy of 81.51 percent. On the other hand, results for LOGO using SVM show that the model is not good at guessing from containers it has not seen before. The accuracy is 6.53 percent.

B. MLP

We implement an MLP using Keras. For the MLP, we used the Adam optimizer and categorical cross-entropy loss. We used four hidden layers and Rectified Linear Unit activation

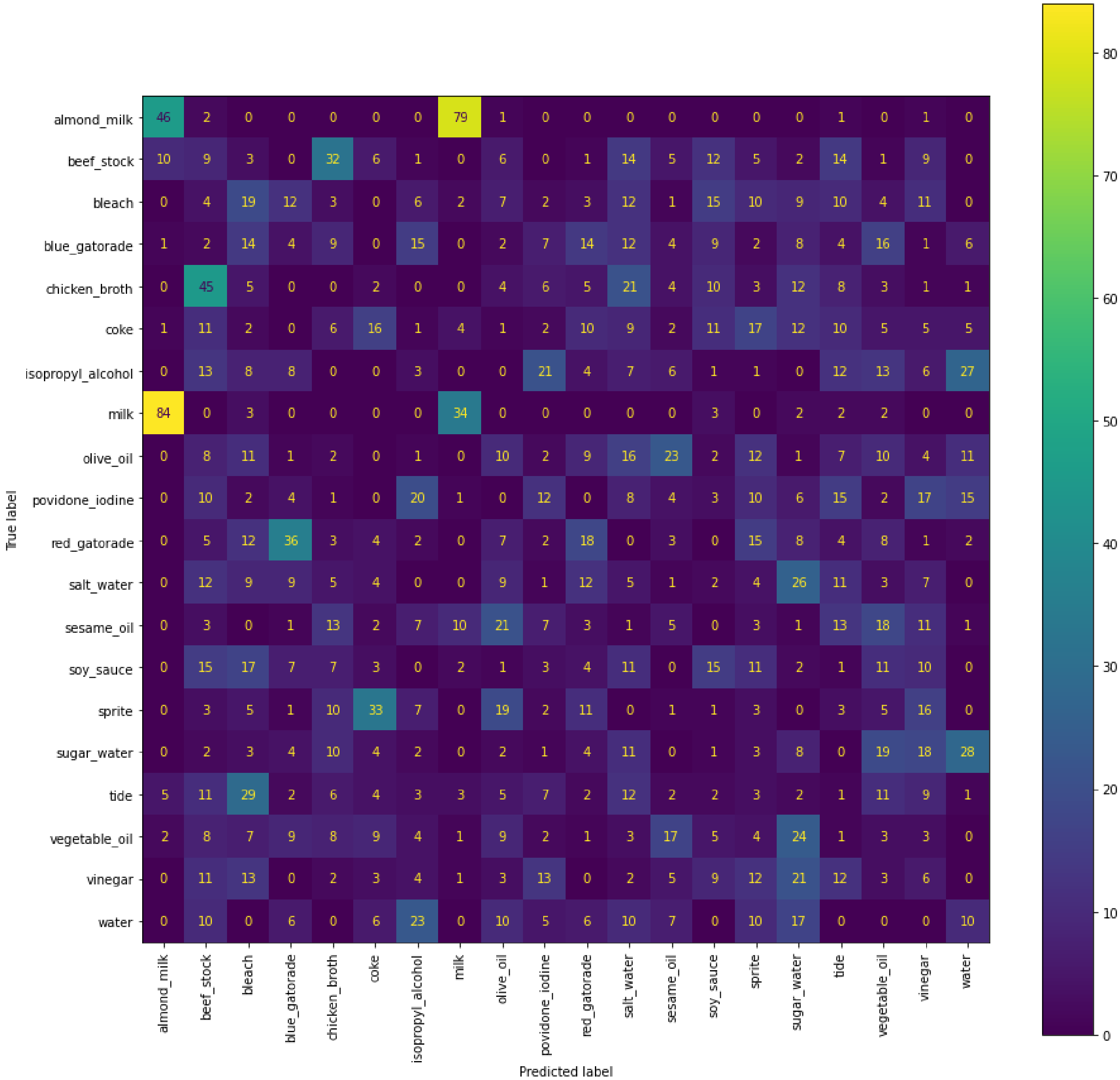


Fig. 6. Confusion matrix for SVM Leave one group out. Only using liquids

for all but the last layer. For the last layer, we used Softmax activation. The number of nodes in each layer decreased such that the first three layers had 5000, 1500, 1000 nodes respectively and the last layer had a single node. The results for Stratified K-fold using MLP and white-black standardization shows the model can estimate containers it has seen before with an accuracy of 55 percent. The MLP with Stratified K-fold verification produced the lowest accuracy. The results for LOGO using MLP and white-black standardization show that the model is not good at generalizing to containers it has not seen before. The accuracy is 2.36 percent. Like the Stratified K-fold accuracy, the MLP LOGO accuracy is also the lowest.

C. *Random Forest*

Lastly, we implement a simple Random Forest using Scikit-learn's. The results for Stratified K-fold using Random Forest and white-black standardization shows the model can estimate containers it has seen before with an accuracy of 78.80 percent. On the other hand, results for LOGO using Random Forest and white-black standardization show that the model is not good at guessing from containers it has not seen before. The accuracy is 9.13 percent.

V. FUTURE WORK

For future works, we first plan on further standardizing our data collection phase by building an enclosure for taking measurements in to reduce the impact of ambient light. We

< TABLE OF CONTENTS

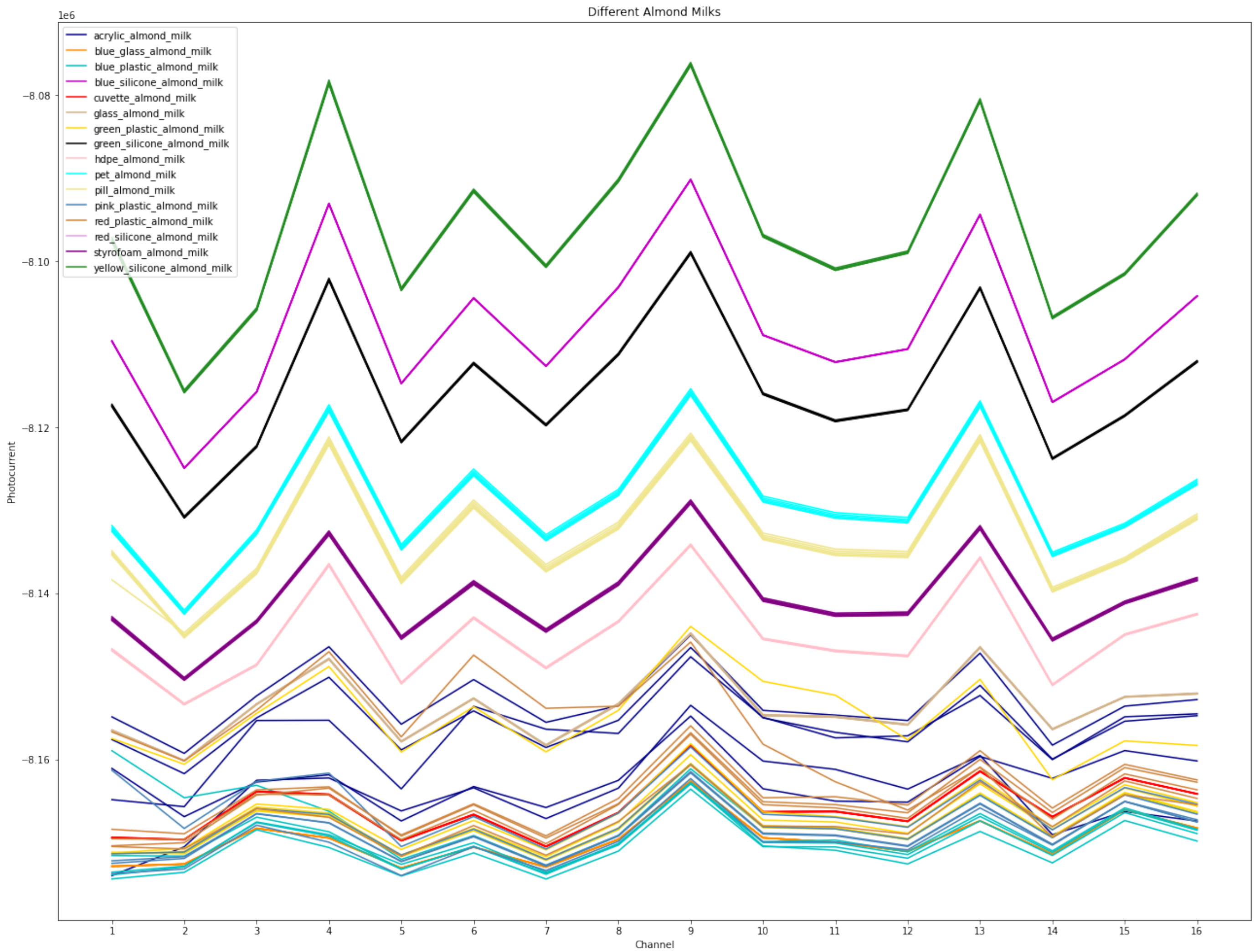


Fig. 7. Plot of almond milk using all containers

Model	Stratified K-fold	Leave-One-Group-Out
SVM	81.51%	6.53%
MLP	55%	2.36%
Random Forest	78.80%	9.13%

Fig. 8. Models (SVM, MLP, Random Forest) and Accuracies: Stratified K-Fold and Leave-One-Group-Out

expect these changes to improve our current signal-to-noise ratio. We also plan on introducing a second spectrometer to increase the electromagnetic sensing range we collect data from. We plan to introduce the Hamamatsu C12880MA, which has a spectral range of 340 to 850 nm, covering the some spectra of visible light not captured by the Spectrapod which has a sensing range of 850 to 1700 nm. Lastly, We plan on having a real-world demonstration where the spectrometers will be integrated onto the end effect of a Stretch RE1 mobile manipulator from Hello Robot.

VI. CONCLUSION

This paper investigates whether NIR Spectroscopy can be used to classify substrates in a way that is invariant to the substrates in containers. We present a dataset of spectral measurements from different combinations of 36 substrates and 16 containers. We train three classifiers to investigate whether we can estimate substrates between previously seen containers and whether those same classifiers can generalize to new, unseen containers. Evaluating our classifiers using Stratified K-fold Cross Validation, We find that it is possible to estimate across containers, with SVM achieving the highest accuracy. However, evaluation via Leave-One-Group-

Out Cross Validation shows that generalization is not yet promising, indicating the need for future work towards this goal.

References

[1] Z. Erickson, N. Luskey, S. Chernova, and C. C. Kemp, "Classification of household materials via spectroscopy," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 700–707, 2019.

[2] L. Bokobza, "Near infrared spectroscopy," *Journal of Near Infrared Spectroscopy*, vol. 6, no. 1, pp. 3–17, 1998. [Online]. Available: https://doi.org/10.1255/jnirs.116

[3] C. Pasquini, "Near infrared spectroscopy: a mature analytical technique with new perspectives – a review," *Analytica Chimica Acta*, vol. 1026, 04 2018.

[4] Z. Erickson, E. Xing, B. Srirangam, S. Chernova, and C. C. Kemp, "Multimodal material classification for robots using spectroscopy and high resolution texture imaging," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 10 452–10 459.

[5] K. Hakkel, M. Petruzzella, F. Ou, A. Klinken, F. Pagliano, T. Liu, P. Veldhoven, and A. Fiore, "Integrated near-infrared spectral sensing," *Nature Communications*, vol. 13, 01 2022.

[6] R. A. Crocombe, "Portable spectroscopy," *Appl. Spectrosc.*, vol. 72, no. 12, pp. 1701–1751, Dec 2018. [Online]. Available: http://opg.optica.org/as/abstract.cfm?URI=as-72-12-1701

[7] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/12/2762

[8] H.-J. Huang, X. Guo, and W. Yuan, "Understanding dynamic tactile sensing for liquid property estimation," 2022. [Online]. Available: https://arxiv.org/abs/2205.08771

Optimality vs. Efficiency in Globally Constrained Path Planning Problems

William N. Scott[1]

***Abstract*— Within the field of robotics, various algorithms have emerged to compute the optimal solution for the shortest-path problem (SPP). We are motivated to use these algorithms to address the growing issue of navigation in congested settlements within South Africa. These communities are densely populated with minimal critical infrastructure for municipal services such as roads or electricity. Consequently, emergencies like fires or floods cause extensive damage to these communities. To better assess these algorithms' efficiency and optimality to solve globally constrained path-planning problems, we test three existing algorithms to simulate their behavior in urban environments. All algorithms extend the basic A* search to compute the optimal path by minimizing an objective function subject to a single global constraint.**

The algorithm's performance was measured separately, with time as a dependent or independent dimension, compared against the Constrained A* (CA*) search developed for re-planning with a global constraint [1]. We achieved optimality in the search when planning in a 2D grid world with time as an independent variable. Our findings communicate that a simplistic A* search planning in (x,y) would serve as the most efficient algorithm to realize urban planning with a constraint, setting up a discussion for efficiency over optimality in path-planning problems with a global constraint.

***Index Terms*— Motion and Path Planning**
Task and Motion Planning

I. INTRODUCTION

The motivation for our work relates to the growing issue of rapid urbanization in the expanding communities of South Africa. To address the immediate demand for shelter, numerous communities establish housing structures independent of the government. These communities lack sufficient infrastructure for municipal services, compounding the damages done by emergencies such as fires, floods, assaults, and illness. We can model the issue of traversing these communities to deliver crucial resources as a shortest-path problem: with a singular agent attempting to navigate a cluttered environment to reach a goal state subject to a global constraint.

The shortest path problem (SPP) in robotics exists with several variations that require an optimal solution. The optimization issue can be assumed as minimizing or maximizing a cost or objective function, such as distance traveled or time elapsed [1]. We often require these optimal solutions to conform to a specific local or global constraint unique to our problem definition. We define a local constraint as an operation able to be computed within a singular step of the solution. In contrast, a global constraint must be evaluated over the entire solution to satisfy the restriction. We can evaluate local constraints with minimal overhead or increased complexity. However, the typical approach for evaluating global constraints entails the addition of another variable/dimension to track each constraint, ending the current search iteration upon a constraint violation. Consequently, the introduction of one or multiple global constraints can rapidly expand the complexity of our search.

Fig 1. highlights the effect of a global constraint on our solution path. *Fig. 1 (a)* shows the optimal solution for the given culdesac map with a global-constraint limiting the path size to a maximum of 70 steps. As we continually increase our global constraint, demonstrated in *Fig. 1 (b)* and *Fig. 1 (c)*, the solution path follows the innate behavior of the A* search to compute the least-cost route, indicated by dark blue regions on the map. This innate behavior relates to the principle concept of the A* search of expanding the next minimum cost state on the graph.

II. BACKGROUND

A. Related Work

We accredit Logan's ABC algorithm as one of the first complete and formal assessments of this problem [2]. The ABC algorithm exists as a generalized A* search with the capability to manage multiple global constraints, but only a minimal set of constraints return an optimal solution.

During the creation of the re-planner Constrained D* (CD*) algorithm, Anthony Stentz developed the preliminary Constrained A* (CA*) algorithm [1]. This more basic version of CD* calculates the most efficient path once, subject to a constraint. The subject of our focus will be the CA* algorithm since we wish to minimize computational power and retrieve an effective solution expeditiously.

B. Variable Dependence

To better conceptualize the problem and our testing methodology, we identify the difference between independent and dependent variables in search algorithms. Independent variables are the criteria we use to differentiate between the states, or grid cells, on a map. In a discrete environment, we can choose to categorize a cell only through its position about the *x* and *y-axis*. Alternatively, we adopt an implementation that specifies a cell with relation to its *x* and *y* coordinates and time *t*, representing the timestamp for when a robot visits a location. Dependent variables are spaces of memory allocated outside the state definition to help compute different metrics unique to a given path planning problem.

[1]William Scott is a Senior Computer Engineering Student interning with The Robotics Institute as a R.I.S.S. scholar at Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA. wscott2@andrew.cmu.edu

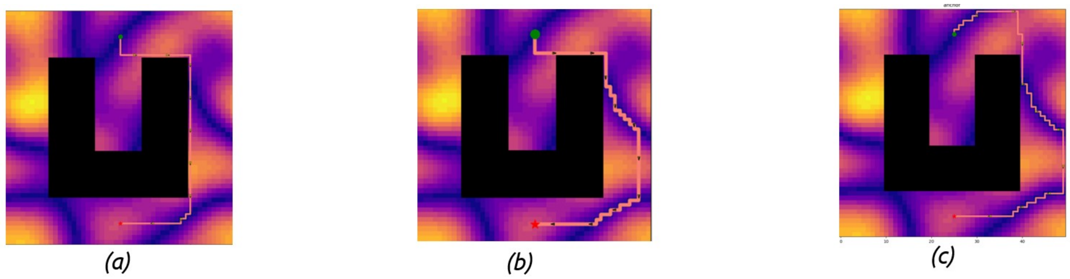


Fig. 1. A*: Increasing Global Constraint

These variables typically assist with evaluating cost metrics or pruning future potential cells for the robot to visit.

III. PROBLEM STATEMENT

A. A* Search with Global Constraint

We venture to derive a mathematical representation of our path planning problem with a global constraint. Consider a cell, *s*, defined with independent variables *(x,y,t)* corresponding to the orientation of the cell in 2D space, and the time our robot views the cell, respectively. Let Π represent any path between two arbitrary cells in our grid space.

$$\Pi = \{s_{i-2}, ..., s_i\} \tag{1}$$

Let Π_X represent the optimal solution from our start position, denoted by s_{start} , to the goal state, s_{goal}, via a set of cells.

$$\Pi_X = \{s_{start}, ..., s_{goal}\} \tag{2}$$

Let $c(\Pi_X)$ denote the cost function we wish to optimize in our A* search where $c(\Pi_{i-1}, \Pi_i)$ represents the action cost for the robot to traverse to an arbitrary cell on the path, Π_i, from any valid previous position, Π_{i-1}, on a 4-connected grid. Let us assume $c(\Pi)$ is optimized when minimizing the overall cost, measured by the total accrued cost by the robot traversing the optimal path $c(\Pi_X)$. Finally, let *n* represent the number of steps in a given path with *T* as the global constraint limiting the path size. We introduce a global constraint into our calculations by optimizing our cost function subject to an value of *n* within our budget *T*.

$$c(\Pi_X) = \Sigma_i^n c(\Pi_{i-1}, \Pi_i) \quad S.T \quad n \leq T \tag{3}$$

B. Constrained A* Search

At the algorithm's core, CA* operates identical to an A* search with dependent time in reference to the cell definition, *s*, and management of the *OPEN* list. Let f_0 represent the objective function to be optimized, synonymous with $c(\Pi_X)$ above. Let f_1 denote the global constraint imposed onto the problem, assume the same limit on the path length as before, $n \leq T$. CA* attempts to find an optimal solution by minimizing a composite function, $f(\Pi, w)$ composed of f_0, f_1, and an adjustable scalar value *w* [1].

$$f(\Pi, w) = f_0 + w * f_1 \tag{4}$$

The algorithm performs a binary search over the weight *w*, from minimum and maximum values provided by the user, to find a *w* value that is just large enough to satisfies the global constraint.

IV. TESTING METHODOLOGY

Three closely related algorithms were realized in our research: two variations of the A* search — time-dependent and independent — and the CA* algorithm. These algorithms were implemented with hash tables in C++ to allow for memory management and ensure computational efficiency. All implicit A* searches performed by the various algorithms utilized a reserve Dijkstra search as the underlying heuristic function to help approximate the cost of the least-cost path to the goal state. Cost refers to an expenditure of time, energy, or resources the robot uses on its journey. We interpret yellow regions on the map as "high cost" areas, denoting an increased difficulty to traverse the area. The dark blue regions represent the lowest cost regions on the map, with purple indicating a median value. In our context, we interpret these costs as the difficulty for an emergency responder to traverse a given space due to debris or any stationary obstacle not absolutely obstructing the agents' movement.

Furthermore, we ran the algorithms on 2D grid maps modeling urban environments: Berlin, Boston, and Paris, from Nathan Sturtevant's MovingAI benchmark database. We chose these maps to test the algorithms' performance across multiple environments to precisely gauge their baseline performance. These maps were processed through a first-order Gaussian filter to randomly assign cost values to every individual cell.

V. RESULTS

We began with realizing a rudimentary A* search in *(x,y,t)*. This A* search operates with time as an independent variable, pruning all sub-optimal paths by discarding cells previously visited by the algorithm later in time. In conjunction with A*'s minimum heuristic check, this algorithm guarantees optimality and completeness. This algorithm will serve as the baseline to compare the performance of the CA* algorithm and A* search with time measured dependently.

In reference to *Table I*, we see the run-time of the least-cost path vastly out performs the run-time of the shortest path solution, realized through a global constraint on the path length. We expect this behavior as the Dijkstra heuristic we

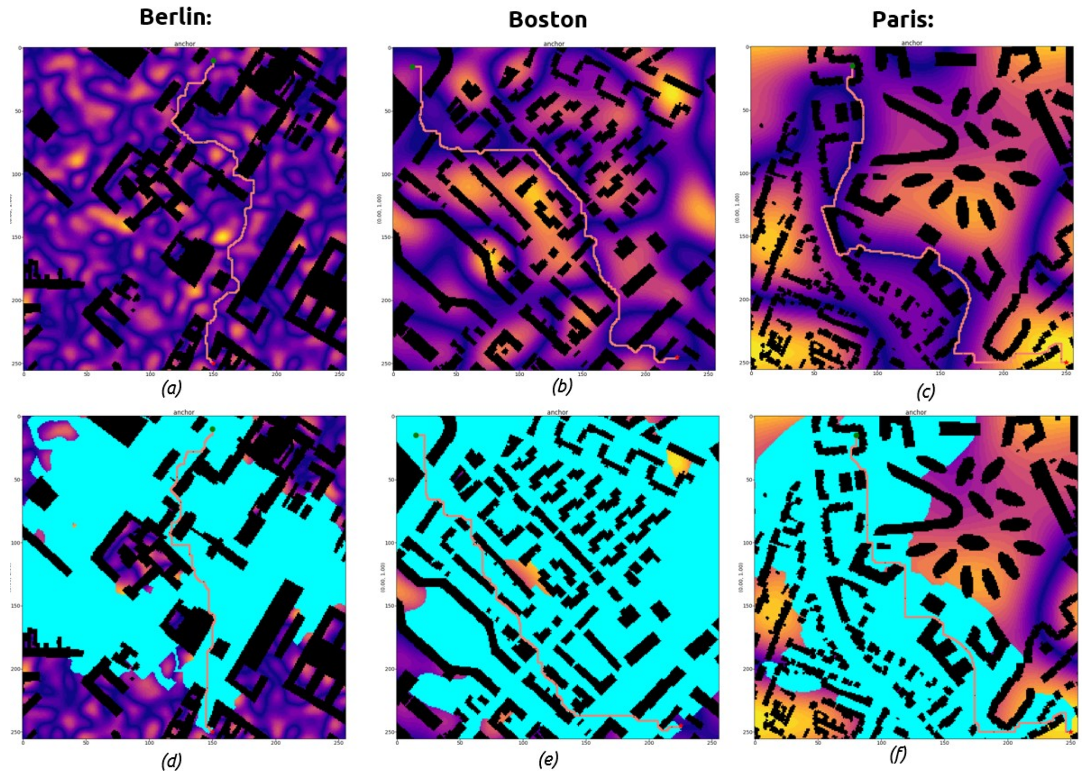


Fig. 2. Least Cost Path vs. Shortest Path in Urban Environment

TABLE I
A*: INDEPENDENT TIME

Map	Solution Path	Run-Time (sec)	Path Length	Path Cost	Cells Expanded
Berlin	*least-cost*	.00653	409	9750	411
Berlin	*shortest*	19.25	338	13890	2091488
Boston	*least-cost*	.0068	489	14730	491
Boston	*shortest*	36.41	448	25300	3884688
Paris	*least-cost*	.0079	566	32710	568
Paris	*shortest*	33.41	443	41470	3408124

TABLE II
A*: DEPENDENT TIME

Map	Solution Path	Run-Time (sec)	Path Length	Path Cost	Cells Expanded
Berlin	*least-cost*	.00572	409	9750	410
Berlin	*shortest*	.00573	409	9750	410
Boston	*least-cost*	.0062	489	14730	490
Boston	*shortest*	.00598	489	14730	490
Paris	*least-cost*	.00692	566	32710	567
Paris	*shortest*	.00681	566	32710	567

use provides the precise cost to transition from our goal state to any cell on the graph. Consequently, the algorithm need only to expand the cells along the least-cost path in order to reach the goal state, shown in *Fig. 2(a,b,c)* by the light blue line underneath the solution path. We highlight the difference between the computation efforts to calculate the least-cost path against the shortest-path in *Fig. 2 (d,e,f)*. These figures provide a visualization of all the cells expanded by the A* algorithm to find all the potential solutions subject to our global constraint. The quantitative number of expansions are provided in column six of *Tables I, II,* and *III* respectively.

The next stage consisted of modifying our existing A* search to operate with time as a dependent variable, reference *Table II*. We developed an A* search with dependent time as an intermediate step to build towards the CA* algorithm. We must acknowledge our environment does not consist of any dynamic obstacles, so the difference between an A* with time measured independently versus dependently will be negligible. Our findings reaffirm this notion as the respective solution paths between the two algorithms were found to be identical. However, by planning exclusively in *(x,y)-space* we reduce the dimensions of the algorithm by one, greatly improving the run-time of our shortest path solution across all maps. In this exchange for faster performance, the A* search with dependent time would not ensure completeness given the presence of dynamic obstacles.

We realized the Constrained A* algorithm as the final stage of our research, in reference to *Table III*. In order to

< TABLE OF CONTENTS

TABLE III

CONSTRAINED A*

Map	Solution Path	Run-Time (sec)	Path Length	Path Cost	Cells Ex-panded	W
Berlin	*least-cost*	2.49	409	9750	410	0
Berlin	*shortest*	15.84	339	23149	22935	37.49
Boston	*least-cost*	2.67	489	14730	490	0
Boston	*shortest*	23.19	447	313563	46293	664.76
Paris	*least-cost*	1.47	566	32710	567	0
Paris	*shortest*	22.11	422	114819	44163	179.9

calculate the least-cost path, CA* must complete multiple calls to the embedded A* algorithm to allow the binary search to reach the minimum w of zero, corresponding to optimizing only f_0 in (4). In contrast, the binary search over W decreases run-time for the shortest path calculation in relation to our A* search with independent time.

VI. CONCLUSIONS & FUTURE WORK

Our findings communicate a standard A* search planning only in *(x,y)-space* would be the most time efficient algorithm for planning in an urban environment. We observed the binary search for the optimal W value in the CA* algorithm significantly impairs the speed of our search. We understand the search for the most optimal and complete solution with an A* algorithm planning in *(x,y,t)* drastically increases our run-time with the introduction of a global constraint.

Future work in this field relates to overcoming the challenge of incorporating multiple path constraints within a singular objective function. Perez-Berquist and Stentz suggest a promising approximation algorithm, K2, to address path-planning problems with multiple global or local constraints [3].

ACKNOWLEDGMENT

This work was supported by the Carnegie Mellon University Robotics Institute Summer Scholars (RISS) Program. A special thanks to Rachel Burcin and Dr. John Dolan for providing tremendous academic and mental-health support to the cohort this summer, expanding the robotics community to individuals of all backgrounds. I would like to thank my mentors Dr. Maxim Likhachev and Dhruv Saxena from the Search-Based Planning Lab for their continued assistance with this project. These individuals has progressed my critical thinking skills and analysis of algorithm behavior beyond any level imaginable for myself. Thank you everyone for making this summer a worthwhile academic and social experience.

REFERENCES

[1] A. Stentz, "Cd*: A real-time resolution optimal re-planner for globally constrained problems," in *Plastics*, 2nd ed. Pittsburgh: AAAI, 2002, vol. 3, pp. 605–611.

[2] B. Logan and N. Alechina, *A* with Bounded Costs.* Birmingham , UK: AAAI, 1998, pp. 123–135.

[3] A. Perez-Bergquist and A. Stentz, "K2: an efficient approximation algorithm for globally and locally multiply-constrained planning problems," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 2047–2053.

Evaluation of AutoML Systems on OpenML Binary-Classification Tasks

Xinchen Yang[1], Jieshi Chen[2] and Artur Dubrawski[2]

***Abstract*— The Automated Machine Learning (AutoML) System has powered data scientists and domain experts with its efficient model discovery capabilities and helped address shortages of qualified data scientists. Widely used AutoML Systems include the Auto^{n}ML developed by the Auton Lab at Carnegie Mellon University (CMU), H_2O AutoML, Auto-Sklearn, etc. Various performance evaluations have been done on these AutoML systems. However, as AutoML systems get updated with improved model searching ability and newly added functionalities, we need to obtain a new map to depict the performance of the AutoML systems. Motivated by this goal, we conducted experiments to evaluate the performance of 4 popular AutoML systems, including Auto^{n}ML, H_2O AutoML, TPOT and AutoGluon, on 177 OpenML binary-classification tasks, using Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve as the evaluation metric. We analyzed the experimental data from various aspects, including the relative rankings of the AutoML systems, training-testing performance discrepancies, relationship between the performance of AutoML systems and dataset characteristics, and the winning algorithms used by each AutoML system.**

***Index Terms*— AutoML, OpenML, model selection**

I. INTRODUCTION

An automated machine learning (AutoML) pipeline is a combination of a series of steps of data preprocessing, feature selection, model and parameter tuning, etc. Each pipeline can be regarded as a well-defined classifier or regressor that takes machine leaning (ML) tasks as input and yield the predication values as the output. An AutoML system has many such pipelines in its search space. When given a ML task, an AutoML system will look for pipelines in its search space, apply these pipelines to the task and rank the performance of the pipelines according to a selected metric (accuracy, AUC, etc). We want to know whether a designed AutoML system can find the best pipelines given a ML task. In this sense, an AutoML system can be regarded as a ”huge” machine learning model. For each task given to an AutoML system, its performance is represented by the performance of the best pipeline it returns —- the pipeline with the highest evaluation score on the AutoML's leaderboard of its searched pipelines.

Multiple open-source AutoML systems are available to use now and they are evolving rapidly. Here is a list of a few of them:

(1) Auto^{n}ML: Auto^{n}ML is an open-source AutoML system developed by CMU Auton Lab [1] using DARPA D3M ecosystem, aiming at power data scientists with efficient model discovery and advanced data analytics. Auto^{n}ML takes the training data as input, then conducts several operations including featurization, fitting and prediction, and validation. Auto^{n}ML outputs a leaderboard of ranked pipelines and the pipelines on the leaderboard can be used to make predictions on the testing data.

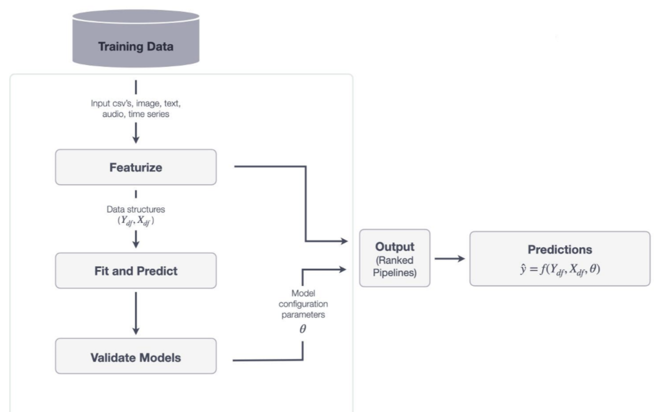


Fig. 1. Auto^{n}ML workflow

(2) H_2O AutoML: Presented by LeDell and Poirier [2], H_2O AutoML is an open-source, highly scalable, fully-automated AutoML framework. H2O AutoML uses a combination of fast random search and stacked ensembles to achieve competitive results. H_2O AutoML has an easy-to-use interface by providing simple wrapper functions that perform a large number of modeling tasks in order to save time for the user.

(3) Tree-Based Pipeline Optimization Tool (TPOT): TPOT is an open-source genetic programming-based AutoML framework introduced by Olson and Moore [3]. The goal of TPOT is to automate the pipeline building process by combining tree representation of pipelines with stochastic search algorithms. TPOT makes use of the Python-based scikit-learn library. .

(4) AutoGluon-Tabular: AutoGluon-Tabular is an open-source AutoML framework that utilizes the technique of ensembling multiple models and stacking them in multiple layers presented by Erickson [4]. The multi-layer combination of many models makes AutoGluon an AutoML that can produce results very quickly with still good results, which can serve the practical uses well.

In this paper, we evaluated the predication performance of 4 AutoML systems: Auto^{n}ML, H_2O AutoML, TPOT and AutoGluon on 177 OpenML binary-classification tasks. Our

[1]Xinchen Yang is with the Department of Electrical and Computer Engineering, New York University, 6 MetroTech Center, Brooklyn, NY, USA xy2332@nyu.edu

[2]Jieshi Chen and Artur Dubrawski, PhD, are with the Auton Lab, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA jieshic@andrew.cmu.edu, awd@cs.cmu.edu

goal is to compare the relative performance of these AutoML systems, identity whether there are relationships between the performance of AutoML systems and certain factors such as time budget and datasets characteristics. We also would like to know whether the top pipelines of those AutoML systems towards the same dataset are different in their selection of core algorithms or not.

II. RELATED WORK

Research has been conducted on the evaluation of AutoML methods and frameworks. Many developers of the existing AutoML systems conducted an evaluation on their AutoML system against other AutoML systems when they introduced their works. For example, the inventors of H_2O AutoML, LeDell and Poirier, evaluated H_2O AutoML against several other AutoML systems on the OpenML AutoML benchmark, which contains 44 classification tasks, proving the effectiveness of the H_2O AutoML [2]. Olson and Moore introduced TPOT and benchmarked TPOT's performance on a set of 150 supervised classification tasks and found that it significantly outperforms a basic machine learning algorithm in 21 tasks and has a minimal degradation only on 4 tasks. It is all accomplished without domain knowledge or tedious manual efforts, which shows a great promise of GP-based AutoML systems [3]. In addition, Erickson, who presented AutoGluon-Tabular, evaluated AutoGluon-Tabular on a suite of 50 classification and regression tasks from Kaggle and the OpenML AutoML Benchmark against several other AutoML platforms, showing the robustness and high performance of AutoGluon-Tabular. AutoGluon-Tabular was also showed to be a time-saving AutoML system in their experiment compared to others [4].

Comprehensive surveys have also been done on the evaluation of the AutoML systems. For example, Zoller and Huber evaluated the performance of a set of methods of algorithm selection and hyperparametr optimization and 6 AutoML frameworks (TPOT, Hyperopt-sklearn, Auto-Sklearn, Random Search, ATM, H_2O AutoML) on 137 OpenML datasets [5]. Ferreira conducted empirical evaluations of 8 AutoML tools on 12 OpenML datasets and compared the best scores achieved by the AutoML tools with the best OpenML public results, confirming the potential of AutoML tools to fully automate the manual efforts on model selection and hyperparameter tuning [6]. Truong evaluated a selected subset of AutoML tools on nearly 300 OpenML datasets, observing that most AutoML tools are able to obtain reasonable results in terms of their performance across many datasets, but there is no "perfect" tool that can outperform all others on a plurality of tasks yet [7].

Tools and platforms have been introduced to facilitate the evaluation and analysis of the AutoML frameworks. For example, Milutinovic introduced an standardized, open-source machine learning framework, D3M, upon which AutoML systems can be evaluated with their strengths and weaknesses exposed. Milutinovic also demonstrated the viability of the D3M framework through the evaluations of 8 AutoML systems upon it [8]. The Auto^{n}ML system developed by the Auton Lab at Carnegie Mellon University (CMU) is built upon the D3M framework.

III. METHODS

Experiments are set up to answer the following questions:

(1) What is the performance difference of the AutoML systems compared to each other?

(2) What is the performance difference of the AutoML systems under different time budgets?

(3) What is the performance difference of the AutoML systems evaluated on datasets with different characteristics?

(4) What is the main contributor to the discrepancies between the performance of AutoML systems, core algorithm selection or other factors?

To answer the questions above, we evaluated 4 AutoML systems: Auto^{n}ML, H_2O AutoML, TPOT and AutoGluon, on 177 OpenML binary classification tasks. Datasets are selected with varied dimensionalities and number of instances. Experiments were run separately on 3 machines, tagged as "Lab Server", "Desktop" and "NYU". All of them are 8-core Linux Machines. First, each dataset is randomly split into training and testing data, with 75% of the original data used as the training data and 25% used as the testing data. Next, the same training and testing data on each task are passed as input to each of the AutoML systems. The metric used to measure the performance of the AutoML systems is AUC. Each AutoML system will look for plausible machine learning pipelines in its search space and rank them according to the training score in AUC. For each AutoML system, the Top 1 pipeline on its leaderboard is used to predict on the testing data, representing the AutoML system that finds it. We set up 3 experimental groups with the time budget to be 60 seconds, 600 seconds and 1200 seconds respectively. The same evaluation process is repeated within the 3 groups only with altered time budget.

IV. RESULTS

After running the experiments, we collected the train and test prediction score of the AutoML systems on the 177 datasets under 3 different time budgets and analyzed the experimental data from several aspects.

A. *Performance of AutoML Systems over Time Budget*

To illustrate the performance of AutoML systems over different time budget, we compared the relative rankings of the performance of the AutoML systems on the test data under the time budget of 60 seconds, 600 seconds, and 1200 seconds, as shown in Figure 2. For each data point, its X-coordinate denotes the time budget of the experiment and its Y-coordinate is obtained by averaging the performance rankings of its corresponding AutoML system across all datasets, with corresponding 95% confidence intervals.

In Figure 2, AutoGluon shows to be an AutoML system with good performance, getting the first place under the time budget of 60 seconds and 600 seconds, and getting the second place under the time budget of 1200 seconds. The average rank of Auto^{n}ML is rather stable across different

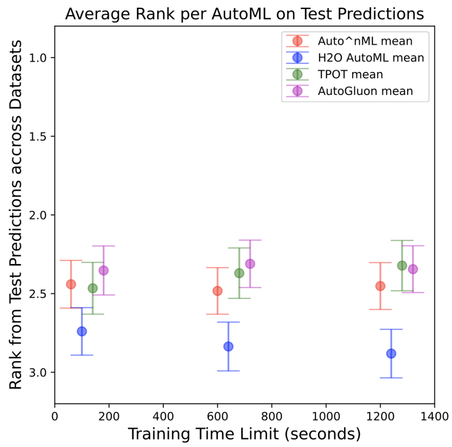


Fig. 2. Average Rank of Test AUCs per AutoML

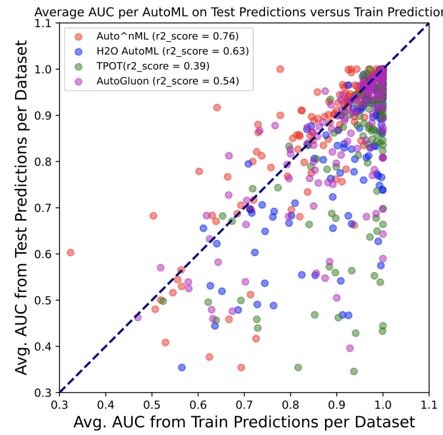


Fig. 3. Average AUC of testing data versus average AUC on Training data per Dataset

time budgets. The average rank of TPOT increases as the time budget increases. In addtion, we can observe that the performance of Auto^{n}ML, TPOT, and AutoGluon are relatively close to each other, while the performance of H_2O AutoML apparently lags behind. As the time budget increases, H_2O AutoML even shows a bigger disadvantage towards other AutoML systems.

TABLE I
RANK STATISTICS ON TEST DATA PER TIME BUDGET, AVERAGED ACROSS N=177 DATASETS, $\pm$ STANDARD ERROR OF THE MEAN

	60 seconds	600 seconds	1200 seconds
Auto^{n}ML	2.44$\pm$0.15	2.48$\pm$0.15	2.45$\pm$0.15
H_2O AutoML	2.74$\pm$0.15	2.84$\pm$0.15	2.88$\pm$0.15
TPOT	2.47$\pm$0.16	2.37$\pm$0.16	2.32$\pm$0.16
AutoGluon	2.35$\pm$0.16	2.31$\pm$0.15	2.34$\pm$0.15

B. *Training-Testing Performance Discrepancies*

To illustrate the relationship between the training and testing performance of the AutoML systems, we make a plot to compare the performance metric (absolute AUC score) from the testing data versus those from the training data as shown in Figure 3. For each data point, its X-coordinate is obtained by averaging the training performance (based on AUC score) of its corresponding AutoML across all time budgets on a specific dataset. Similarly, for each data point, its Y-coordinate is obtained by averaging the test performance (based on AUC score) of its corresponding AutoML across all time budgets on the same dataset. We also include the diagonal line $y = x$ to denote the ideal scenario when the test prediction score is equal to the train prediction score.

We use R-squared (R^2) score as a metric to evaluate the degree of consistency between training AUC scores and testing AUC scores across all datasets per AutoML system. We can observe that Auto^{n}ML, H_2O AutoML, TPOT and AutoGluon achieve a R^2 score of 0.76, 0.63, 0.39, and 0.54 respectively. It shows that, compared to the other AutoML systems, Auto^{n}ML has a stronger correlation between its test prediction score and train prediction score, which is a desired feature. We hope that the data points can get as close to the line $y = x$ as possible, because we want the train performance of the AutoML systems can serve as a good indicator of the test performance of the AutoML systems. Here, we can observe that TPOT have many data points that stray far away from the line $y = x$, with a test performance far worse than the train performance, which indicates that TPOT may suffer from an over-fitting problem on some data tasks.

C. *Relationship Between AutoML System Performances and Dataset Characteristics*

We are interested in how the characteristics of the datasets, in particular, dimensionality and number of instances, affect the performance of the AutoML systems.

To illustrate the relationship between the performance of the AutoML systems and the dimensionality of the datasets, we divide the whole range of dimensionality into distinct "buckets" using an interval of 10. Within each dimensionality "bucket", we average the test prediction scores over all time budgets over all datasets for each AutoML to obtain a data point. The results are shown in Table II and Figure 4.

Similarly, to illustrate the relationship between the performance of the AutoML systems and the number of instances of the datasets, we divide the whole range of number of instances into distinct "buckets" using an interval of 1000. Within each "bucket" of number of instances, we average the test prediction scores over all time budgets over all datasets for each AutoML to obtain a data point. The results are shown in Table III and Figure 5.

Regarding the relationship between the performance of AutoML systems and the dimensionality of datasets, we can see that H_2O AutoML remains in a lagging position almost over all dimensionalities, as shown in Figure 4, which is consistent to its state of falling behind in its average rank across datasets over different time budgets. As the dimensionalities increase, the average rank of Auto^{n}ML climbs up first, reaching its peak when the dimensionality

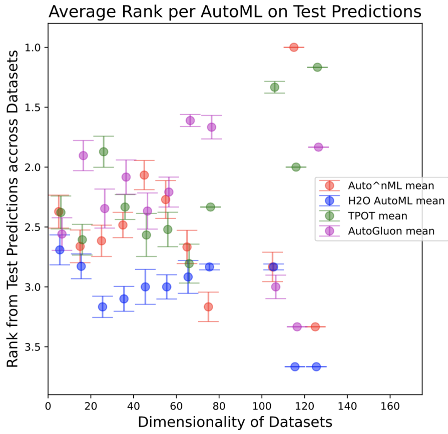


Fig. 4. Average Rank per AutoML on Test Predictions across Datasets within Different Dimensionality Groups

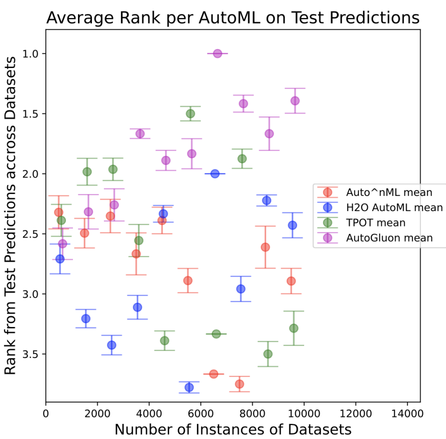


Fig. 5. Average Rank per AutoML on Test Predicitions accross Datasets within Different Groups of Number of Instances

is within the range of [40,50), then falls down, forming the shape of a parabola. However, we are not sure whether this pattern can reveal certain relationships between the performance of Auto^{n}ML and the dimensionality of datasets. We are not certain whether this pattern is representative, either. For TPOT and AutoGluon, no obvious relationships between their performance and the dimensionality of datasets can be found for now.

Regarding the relationship between the performance of AutoML systems and the number of instances, Figure 5 shows that AutoGluon has an upward trend in terms of its relative ranking as the number of instances increases. When the the number of instances is greater than or equal to 6000, AutoGluon apparently prevails over other AutoML systems. TPOT, on the other hand, achieves high relative ranks when the dataset is small (e.g. with number of instances less than 3000), but suffers from a significant drop in terms of its relative ranking on the 17 datasets with more than 8000 instances. No definitive conclusion has been found on the relationship between the relative performance of Auto^{n}ML and number of instances of the dataset, yet. However, H_2O AutoML seems to have a lift in its raltive ranking from its lagging position when the number of instances of the dataset is greater than or equal to 6000.

TABLE II
AVERAGE RANK PER AUTOML ON TEST PREDICTIONS ACROSS N=177 DATASETS WITHIN DIFFERENT DIMENSIONALITY GROUPS

Dimensionality	Number of Tasks	Auto^{n}ML	H_2O AutoML	TPOT	AutoGluon
[0,10)	96	2.37±0.14	2.69±0.12	2.38±0.14	2.56±0.14
[10,20)	33	2.66±0.14	2.83±0.10	2.61±0.13	1.90±0.13
[20,30)	13	2.62±0.13	3.17±0.09	1.87±0.13	2.35±0.16
[30,40)	10	2.48±0.11	3.10±0.10	2.33±0.10	2.08±0.14
[40,50)	5	2.07±0.12	3.00±0.15	2.57±0.18	2.37±0.15
[50,60)	8	2.27±0.16	3.00±0.10	2.52±0.14	2.21±0.13
[60,70)	6	2.67±0.14	2.92±0.14	2.81±0.16	1.61±0.05
[70,80)	2	3.17±0.12	2.83±0.02	2.33±0.00	1.67±0.10
[80,90)	0	NaN	NaN	NaN	NaN
[90,100)	0	NaN	NaN	NaN	NaN
[100,110)	2	2.83±0.12	2.83±0.02	1.33±0.05	3.00±0.10
[110,120)	1	1.00±0.00	3.67±0.00	2.00±0.00	3.33±0.00
[120,∞)	1	1.67±0.00	3.67±0.00	1.17±0.00	1.83±0.00

TABLE III
AVERAGE RANK PER AUTOML ON TEST PREDICTIONS ACROSS N=177 DATASETS WITHIN DIFFERENT GROUPS OF NUMBER OF INSTANCES

Number of Instances	Number of Tasks	Auto^{n}ML	H_2O AutoML	TPOT	AutoGluon
[0,1000)	107	2.32±0.14	2.71±0.12	2.39±0.13	2.58±0.13
[1000,2000)	30	2.49±0.12	3.21±0.08	1.98±0.11	2.32±0.14
[2000,3000)	9	2.35±0.14	3.43±0.08	1.96±0.09	2.26±0.13
[3000,4000)	3	2.67±0.17	3.11±0.10	2.56±0.13	1.67±0.04
[4000,5000)	3	2.39±0.11	2.33±0.07	3.39±0.08	1.89±0.08
[5000,6000)	3	2.89±0.10	3.78±0.05	1.5±0.06	1.83±0.13
[6000,7000)	1	3.67±0.00	2.00±0.00	3.33±0.00	1.00±0.00
[7000,8000)	4	3.75±0.06	2.96±0.10	1.88±0.08	1.42±0.07
[8000,9000)	3	2.61±0.18	2.22±0.05	3.50±0.10	1.67±0.14
[9000,∞)	14	2.89±0.11	2.43±0.10	3.29±0.14	1.39±0.10

D. *Pipeline Algorithm Exploration*

1) Frequency of Winning Algorithms: In order to illustrate the distribution of winning algorithms of each AutoML system, we collect the core algorithm used by the top AutoML on each data task at each time budget. When there is a tie, we treat all the AutoML systems that can achieve the highest test prediction score as the "top" AutoML system. We make bar plots to show the frequency of the first-place algorithms for each AutoML system in Figure 6 and the percentage frequency of the first-place algorithms for each AutoML system in Figure 7.

We can see from the percentage frequency plots that Auto^{n}ML has very stable winning core algorithms over varied time budgets. The same eight core algorithms have ever made Auto^{n}ML win across varied time budgets and their percentage splits across varied time budgets are close to each other. "gradient_boosting" is the algorithm that helps Auto^{n}ML win most, taking a share between 20% and 30% of the winning algorithms of Auto^{n}ML across all time budgets. There are several other noticeable points as well. For example, for H_2O AutoML, it starts with exploring a few algorithms when the time budget is limited, including XGBoost, DeepLearning, and Gradient Boosting Machine (GBM). As the time budget increases, H_2O AutoML explores more types of algorithms and uses them to win. However, as the time budget continues to increase, H_2O AutoML returns to its original choice of algorithms. For TPOT, we can see that it is the AutoML system that has the largest number of distinct winning algorithms (over 10 distinct winning algorithms under any time budget), which suggests that TPOT may try a large number of different algorithms, and some of the algorithms may not be in use by other AutoML systems. This diversity may provide TPOT an edge in its searching of good algorithms and pipelines. Actually, TPOT turns out to be a frequent winning AutoML system in this experiment. Together with AutoGluon, TPOT never falls out of the first two places in terms of winning frequency, and they have significantly more winnings than Auto^{n}ML and H_2O AutoML. For AutoGluon, we can observe that its winning algorithms are very stable across varied time budgets. ”WeightedEnsemble_L2” and ”CatBoost” are the Top 2 most frequent winning algorithms of AutoGluon and each of their shares is significantly larger than the share of any other winning algorithm. ”WeightedEnsemble_L2” is the most frequent winning algorithm under any time budget and its edge over ”CatBoost” in frequency is significant, which is consistent to the fact that AutoGluon uses the technique of ensembling several other models to produce its own prediction models. Together with TPOT, AutoGluon is a frequent winning AutoML system over Auto^{n}ML and H_2O AutoML.

TABLE IV

Absolute Frequency and Percentage Frequency of Winning Algorithm for each AutoML System over Experiments over N=177 datasets (Time Budget = 60 Seconds)

	Auto^{n}ML	H_2O AutoML	TPOT	AutoGluon	Total
gradient_boosting	14 / 24.14%	0	8 / 12.31%	0	22 / 9.61%
extra_trees	13 / 22.41%	1 / 2.50%	7 / 10.77%	0	21 / 9.17%
ada_boost	9 / 15.52%	0	0	0	9 / 3.93%
sgd	0	0	0	0	0
bagging	5 / 8.62%	0	0	0	5 / 2.18%
mlp	5 / 8.62%	0	7 / 10.77%	0	12 / 5.24%
random_forest	4 / 6.90%	0	6 / 9.23%	0	10 / 4.37%
XGBoost	4 / 6.90%	13 / 32.50%	10 / 15.38%	1 / 1.52%	28 / 12.23%
logistic_regression	4 / 6.90%	0	3 / 4.62%	0	7 / 3.06%
DeepLearning	0	8 / 20.00%	2 / 3.08%	2 / 3.04%	12 / 5.24%
GBM	0	14 / 35.00%	8 / 12.31%	0	22 / 9.61%
GLM	0	1 / 2.50%	0	0	1 / 0.44%
DRF	0	2 / 5.00%	0	0	2 / 0.87%
GaussianNB	0	1 / 2.50%	3 / 4.62 %	0	4 / 1.75%
MultinomialNB	0	0	2 / 3.08%	0	2 / 0.87%
BernoulliNB	0	0	0	0	0
XGBClassifier	0	0	1 / 1.54%	0	1 / 0.44%
DecisionTreeClassifier	0	0	2 / 3.08%	0	2 / 0.87%
WeightedEnsemble_L2	0	0	0	43 / 65.15%	43 / 18.78%
LightGBMLarge	0	0	0	0	0%
CatBoost	0	0	0	17 / 25.76%	17 / 7.42%
LightGBM	0	0	0	1 / 1.52%	1 / 0.44%
KNeighborsDist	0	0	6 / 9.23%	0	6 / 2.62%
LightGBMXT	0	0	0	2 / 3.03%	2 / 0.87%
ToTal	58 / 100.00%	40 / 100.00%	65 / 100.00%	66 / 100.00%	229 / 100.00%

TABLE V

Absolute Frequency and Percentage Frequency of Winning Algorithm for each AutoML System over Experiments over N=177 datasets (Time Budget = 600 Seconds)

	Auto^{n}ML	H2O AutoML	TPOT	AutoGluon	Total
gradient_boosting	15 / 26.79%	4 / 9.09%	14 / 20.00%	0	33 / 13.69%
extra_trees	9 / 16.07%	1 / 2.27%	21 / 30.00%	0	31 / 12.86%
ada_boost	5 / 8.93%	0	0	0	5 / 2.07%
sgd	0	0	0	0	0
bagging	5 / 8.93%	0	0	0	5 / 2.07%
mlp	6 / 10.71%	0	10 / 14.29%	0	16 / 6.64%
random_forest	6 / 10.71%	0	4 / 5.71%	0	10 / 4.15%
XGBoost	3 / 5.36%	5 / 11.36%	0	2 / 2.82%	10 / 4.15%
logistic_regression	7 / 12.5%	1 / 2.27%	1 / 1.43%	0	9 / 3.73%
DeepLearning	0	9 / 20.45%	0	4 / 5.63%	13 / 5.39%
GBM	0	22 / 50.00%	0	0	22 / 9.13%
GLM	0	0	0	0	0
DRF	0	0	0	0	0
GaussianNB	0	0	3 / 4.29%	0	3 / 1.24%
MultinomialNB	0	0	1 / 1.43%	0	1 / 0.41%
BernoulliNB	0	0	0	0	0
XGBClassifier	0	0	1 / 1.43%	0	1 / 0.41%
DecisionTreeClassifier	0	1 / 2.27%	3 / 4.29%	0	4 / 1.66%
WeightedEnsemble_L2	0	0	0	43 / 60.56%	43/ 17.84%
LightGBMLarge	0	0	0	0	0
CatBoost	0	0	0	20 / 28.17%	20 / 8.30%
LightGBM	0	0	0	0	0
KNeighborsDist	0	1 / 2.27%	12 / 17.14%	0	13 / 5.39%
LightGBMXT	0	0	0	2 / 2.82%	2 / 0.83%
Total	56 / 100.00%	44 / 100.00%	70 / 100.00%	71 / 100.00%	241 / 100.00%

TABLE VI

Absolute Frequency and Percentage Frequency of Winning Algorithm for each AutoML System over Experiments over N=177 datasets (Time Budget = 1200 Seconds)

AutoML	Auto^{n}ML	H2O AutoML	TPOT	AutoGluon	Total
gradient_boosting	13 / 22.41%	0	17 / 23.29%	0	30 / 12.66%
extra_trees	11 / 18.97%	0	20 / 27.40%	0	31 / 13.08%
ada_boost	8 / 13.79%	0	0	0	8 / 3.38%
sgd	0	0	0	0	0
bagging	5 / 8.62%	0	0	0	5 / 2.11%
mlp	6 / 10.34%	0	11 / 15.07%	0	17 / 7.17%
random_forest	4 / 6.90%	0	6 / 8.22%	0	10 / 4.22%
XGBoost	2 / 3.45%	11 / 26.83%	0	2 / 3.08%	15 / 6.33%
logistic_regression	9 / 15.52%	0	1 / 1.37%	0	10 / 4.22%
DeepLearning	0	10 / 24.39%	0	4 / 6.15%	14 / 5.91%
GBM	0	20 / 48.78%	0	0	20 / 8.44%
GLM	0	0	0	0	0
DRF	0	0	0	0	0
GaussianNB	0	0	3 / 4.11%	0	3 / 1.27%
MultinomialNB	0	0	1 / 1.37%	0	1 / 0.42%
BernoulliNB	0	0	0	0	0
XGBClassifier	0	0	1 / 1.37%	0	1 / 0.42%
DecisionTreeClassifier	0	0	3 / 4.11%	0	3 / 1.27%
WeightedEnsemble_L2	0	0	0	37 / 56.92%	18 / 15.61%
LightGBMLarge	0	0	0	0	0
CatBoost	0	0	0	19 / 29.23%	19 / 8.02%
LightGBM	0	0	0	0	0
KNeighborsDist	0	0	10 / 13.70%	0	10 / 4.22%
LightGBMXT	0	0	0	3 / 4.62%	3 / 1.27%
Total	58 / 100.00%	41 / 100.00%	73 / 100.00%	65 / 100.00%	237 / 100.00%

2) Performance Discrepancies versus Core Algorithm Discrepancies: In order to explore whether the performance discrepancies is contributed by the difference in the selection of the core algorithm or not, we pick up one AutoML system, Auto^{n}ML, into study in particular. We focus on the data tasks that Auto^{n}ML did not win. In order to investigate the algorithm distribution on these tasks, we make heat maps of the algorithms used by the winning AutoML versus the algorithms used by Auto^{n}ML. If there is a tie over the Top 1 AutoML system, we take all of them and their core algorithms as well into consideration. The result is shown in Figure 8. The count in each grid denotes the frequencies that Auto^{n}ML did not get the first place with its corresponding algorithm and another AutoML system got the first place with its corresponding algorithm.

Several observations are as below. There is a prominent grid: ”WeightedEnsemble_L2”-”gradient_bossting”. It has a value of 16, 15, 12 under the time budget of 60 seconds, 600

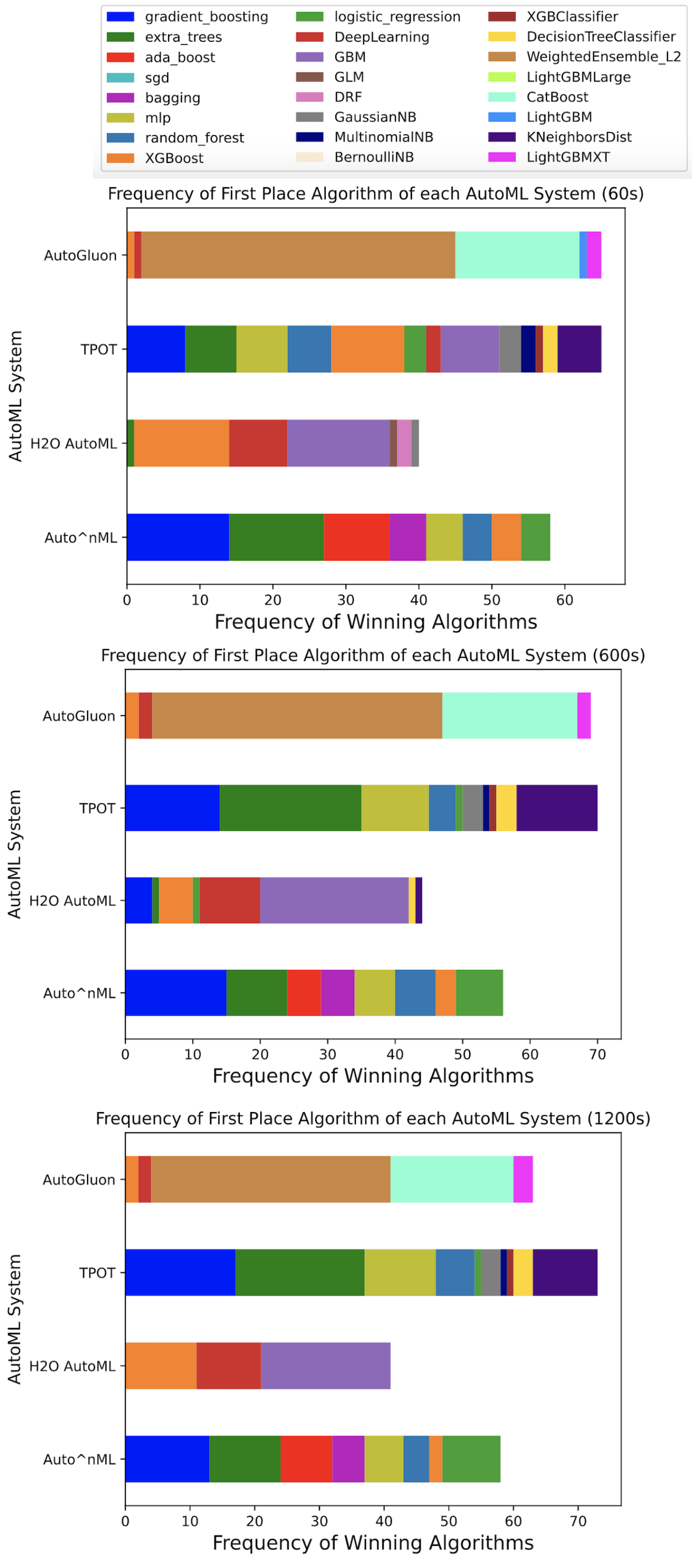


Fig. 6. Frequency of First Place Algorithm of each AutoML

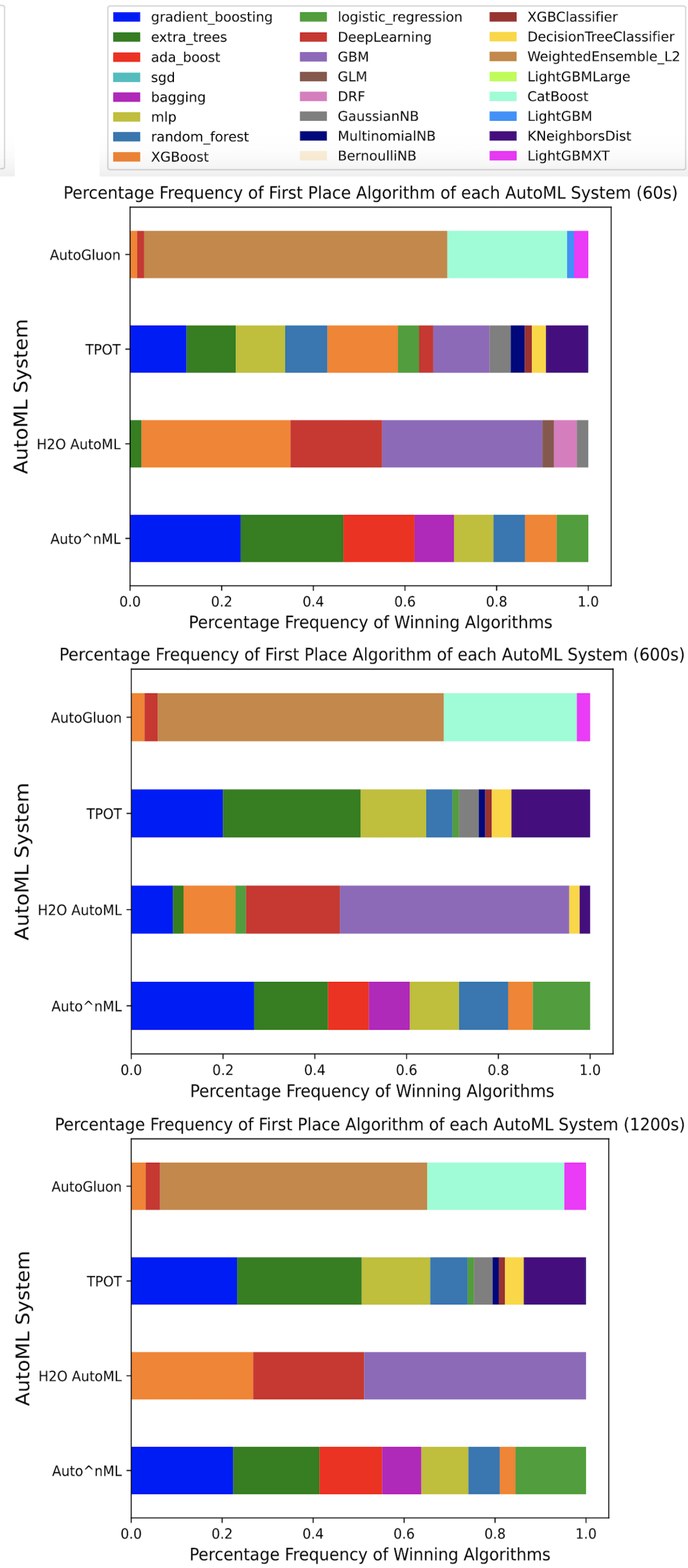


Fig. 7. Percentage Frequency of First Place Algorithm of each AutoML

seconds, and 1200 seconds respectively, which means that there are 16, 15, 12 times when Auto^{n}ML did not win using the algorithm of "gradient_bossting" while another AutoML system won using the algorithm of "WeightedEnsemble_L2" under the time budget of 60 seconds, 600 seconds and 1200 seconds respectively. It shows that Auto^{n}ML has a tendency to utilize the "gradient_boosting" algorithm, but this individual algorithm often failed to beat the ensemble technique. In addition, "WeightedEnsemble_L2" also counts for many times of Auto^{n}ML not getting the first place in total, which are 43, 43 and 37 times respectively under the time budget of 60 seconds, 600 seconds and 1200 seconds. Several other major algorithms that tend to evade Auto^{n}ML as winning algorithms include "DeepLearning", "GBM", "extra_trees" and "gradient_boosting". We noticed that under the time budget of 600 seconds and 1200 seconds, there are 2 additional prominent grids: "extra_trees"-"extra_trees", and "gradient_boosting"-"gradient_boosting". The value of grid "extra_trees"-"extra_trees" are 7 and 5 respectively under the time budget of 600 seconds and 1200 seconds. The value of grid "gradient_boosting"-"gradient_boosting" are 4 and 5 respectively under the time budget of 600 seconds and 1200 seconds. It shows that other AutoML systems may beat Auto^{n}ML by factors other than pipeline core algorithm selection, such as data processing or model parameter selection when the time budget is large.

V. CONCLUSION

We evaluated the performance of 4 AutoML systems on 177 OpenML binary-classification tasks, using AUC as the evaluation metric. We analyzed the experimental data from several aspects, including the relative rankings of the AutoML systems, training-testing performance discrepancies, relationship between the performance of the AutoML systems and the dataset characteristics, and the core algorithms used by each AutoML system that can help it win. We show that AutoGluon achieved the highest average rank among the 4 AutoML systems being evaluated under the time budget of 60 seconds and 600 seconds, while TPOT achieved the highest average rank under the time budget of 1200 seconds. We find that Auto^{n}ML has the strongest correlation between its test prediction score and train prediction score, which indicates better generalization of the predictive models. No definitive conclusion has been found regarding the relationship between the performance of the AutoML systems and dimensionality of the datasets yet. However, there may exist certain relationships between the performance of the AutoML systems and number of instances of the datasets. For example, according to the experimental results on the 177 datasets, the relative performance of AutoGluon goes in an upward trend as the number of instances of the dataset increases, while TPOT generally performs relatively better on smaller datasets than larger ones. Lastly, we showed the absolute frequency and the percentage frequency of winning algorithms of each AutoML system. In addition, we used Auto^{n}ML as an example to study how algorithm selection can contribute to performance discrepancies. We identify

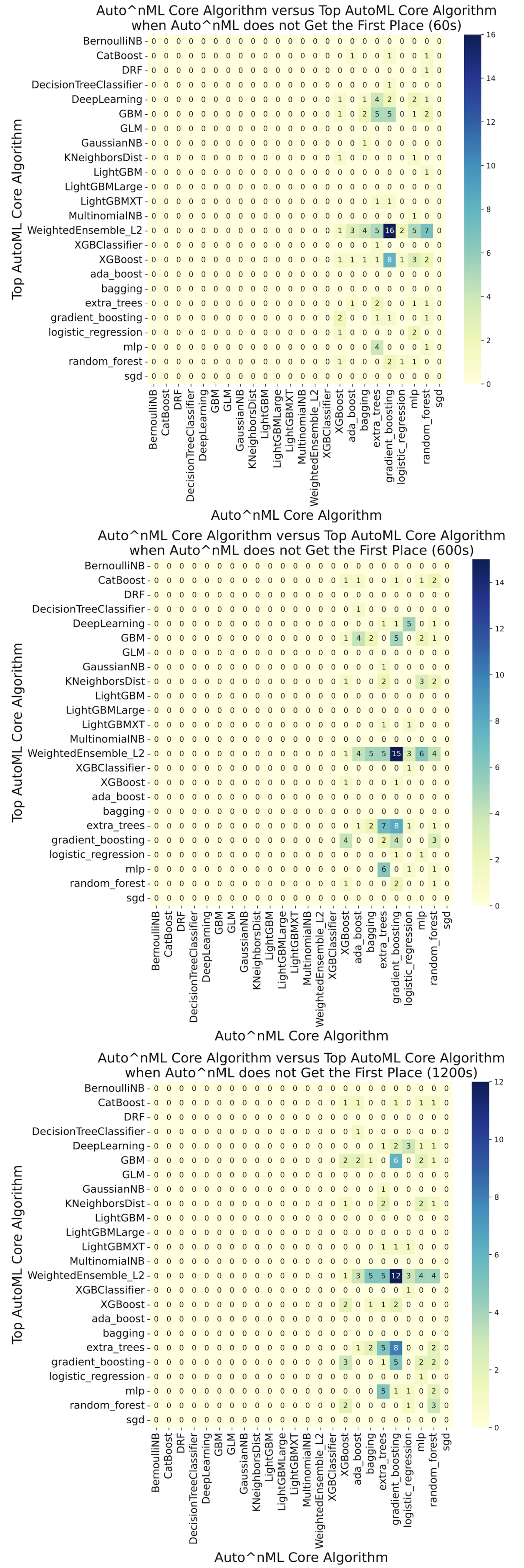


Fig. 8. Heat Maps of Top AutoML Core Algorithm versus Auto^{n}ML Core Algorithm when Auto^{n}ML did not Get the First Place

< TABLE OF CONTENTS

several algorithms that Auto^{n}ML tend to miss but grabbed by other AutoML systems to get the first place, such as the ensemble technique. We showed that both the pipeline algorithm selection and other factors can contribute to the performance discrepancies among the AutoML systems.

However, our research still has some limitations. For example, our selection of data tasks contains too many datasets with small dimensionalities and number of instances and only a few datasets with large dimensionalities or number of instances. Therefore, the performance lift of AutoGluon over large datasets may not be representative. In the future, we hope to incorporate more datasets with large dimensionality or number of instances into our evaluation process. In addition, we solely focus on OpenML binary-classification tasks for now. In the future, we hope to expand our experiment to multi-class classification tasks and regression tasks to see how the AutoML systems behave in these types of tasks. We also like to test AutoML systems on datasets from sources other than OpenML and observe their behaviors over these tasks.

VI. APPENDIX

Please See Table VII attached for summary of average train prediction scores and test prediction scores of the AutoML systems across time budgets. See Table VIII attached for a description of data tasks that the 4 AutoML systems have been evaluated upon.

VII. ACKNOWLEDGMENT

This work is supported by D3M MILEI funding and Carnegie Mellon Robotics Institute Summer Scholars (RISS) program. The authors would like to thank Rachel Burcin and Dr. John M. Dolan for their organization and support of this program.

REFERENCES

[1] "Auto^{n}ml - taking your ml capacity to the nth power," https://autonlab.org/research/usability.html.

[2] E. LeDell, "H2o automl: Scalable automatic machine learning," 2020.

[3] R. S. Olson and J. H. Moore, "Tpot: A tree-based pipeline optimization tool for automating machine learning," in *Proceedings of the Workshop on Automatic Machine Learning*, ser. Proceedings of Machine Learning Research, F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., vol. 64. New York, New York, USA: PMLR, 24 Jun 2016, pp. 66–74.

[4] N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola, "Autogluon-tabular: Robust and accurate automl for structured data," 2020. [Online]. Available: https://arxiv.org/abs/2003.06505

[5] M.-A. Zöller and M. F. Huber, "Benchmark and survey of automated machine learning frameworks," 2019. [Online]. Available: https://arxiv.org/abs/1904.12054

[6] L. Ferreira, A. Pilastri, C. M. Martins, P. M. Pires, and P. Cortez, "A comparison of automl tools for machine learning, deep learning and xgboost," pp. 1–8, 2021.

[7] A. Truong, A. Walters, J. Goodsitt, K. Hines, C. B. Bruss, and R. Farivar, "Towards automated machine learning: Evaluation and comparison of automl approaches and tools," in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, 2019, pp. 1471–1479.

[8] M. Milutinovic, B. Schoenfeld, D. Martinez-Garcia, S. Ray, S. Shah, and D. Yan, "On evaluation of automl systems," in *Proceedings of the ICML Workshop on Automatic Machine Learning*, 2020.

TABLE VII

AVERAGED TRAIN PREDICTION SCORE AND TEST PREDICTION SCORE OF THE AUTOML SYSTEMS ACROSS TIME BUDGETS

	OpenML ID	Auto^{n}ML Train	Auto^{n}ML Test	H_2O Train	H_2O Test	TPOT Train	TPOT Test	AutoGluon Train	AutoGluon Test
0	1013	0.7254	0.4167	0.7647	0.6919	0.9035	0.4621	0.9	0.505
1	823	0.9984	0.9983	0.9999	0.9986	0.9662	0.967	0.9993	0.9987
2	799	0.9634	0.9647	0.9965	0.9654	0.9903	0.9617	0.9735	0.9701
3	1004	1.0	1.0	0.9998	1.0	1.0	0.9992	1.0	1.0
4	842	0.7298	0.88	0.7618	0.68	0.9467	0.8133	0.85	0.8
5	1006	0.9131	0.9662	0.9618	0.9255	0.9943	0.8407	1.0	0.9676
6	737	0.9255	0.9176	0.9548	0.9163	0.9845	0.9333	0.9279	0.9307
7	740	0.9579	0.9786	0.9954	0.9786	0.9984	0.982	0.9751	0.9812
8	1220	0.6927	0.7053	0.7387	0.7062	0.6733	0.6709	0.7055	0.7159
9	757	0.8819	0.8484	0.981	0.8342	0.9997	0.846	0.974	0.8308
10	792	0.9743	0.9858	0.9945	0.9792	0.9974	0.9888	0.9977	0.9841
11	1011	0.9862	0.9846	0.9997	0.9861	0.9963	0.9799	1.0	0.9801
12	803	0.9774	0.982	0.9988	0.9812	0.9922	0.9836	0.979	0.9848
13	13	0.6388	0.6707	0.9321	0.6651	0.7727	0.7434	0.8602	0.7536
14	15	0.9915	0.9977	0.9933	0.9975	0.9967	0.9977	1.0	0.9981
15	37	0.8203	0.8548	0.836	0.8367	0.8941	0.8619	0.8368	0.8747
16	43	0.6639	0.7668	0.7119	0.6966	0.718	0.7412	0.7966	0.7632
17	50	0.9967	0.9997	1.0	1.0	1.0	1.0	1.0	1.0
18	333	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
19	334	0.9998	1.0	1.0	0.9972	1.0	1.0	1.0	1.0
20	335	0.9934	0.9877	0.9987	0.9755	0.9998	0.9936	0.9959	0.9812
21	346	0.7182	0.8	0.6404	0.6222	0.797	0.8333	0.7333	0.8667
22	444	0.7301	0.863	0.9161	0.8296	0.8863	0.9346	0.8889	0.8704
23	448	0.9378	0.6625	0.9986	0.7392	0.9854	0.6775	1.0	0.59
24	450	0.9951	0.9935	0.9678	0.9738	1.0	0.9869	1.0	0.9803
25	451	0.9914	0.9893	0.9965	0.9943	0.9961	0.9979	0.9978	0.994
26	464	0.9221	0.9645	0.9258	0.9655	0.9666	0.9746	0.9583	0.9746
27	472	0.845	0.775	0.9582	0.7597	0.8792	0.7833	1.0	0.8083
28	476	0.9697	1.0	0.996	0.9889	0.9106	0.9	1.0	0.9667
29	479	0.8833	0.9296	0.9853	0.9	0.9455	0.9222	0.9091	0.9333
30	949	0.8371	0.734	0.8098	0.7099	0.967	0.7974	0.8531	0.7814
31	1037	0.8994	0.9108	0.9439	0.9101	0.9161	0.9071	0.9177	0.9095
32	1566	0.8232	0.9103	0.9984	0.9976	1.0	0.9978	0.9989	0.987
33	744	0.9282	0.9465	0.9989	0.9674	1.0	0.9571	0.93	0.93
34	1558	0.9065	0.8795	0.9966	0.8801	0.959	0.8831	0.9181	0.8934
35	1024	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
36	23499	0.7153	0.6951	0.7968	0.6668	0.8403	0.6638	0.9853	0.6873
37	1167	0.6303	0.6833	0.7449	0.649	0.6531	0.6822	0.6182	0.6829
38	1511	0.9627	0.9673	0.9826	0.9551	0.9895	0.9563	0.9778	0.9704
39	1524	0.9155	0.9289	0.9784	0.8958	0.9419	0.9449	0.9415	0.9067
40	890	0.8823	0.9013	0.8869	0.7632	1.0	0.9013	1.0	0.8092
41	1455	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
42	1473	0.5644	0.5298	0.925	0.6825	0.8971	0.5595	0.8846	0.5833
43	1463	0.9202	0.7833	0.969	0.8122	0.9718	0.6366	0.9773	0.77
44	1495	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
45	40714	0.1806	0.1562	0.5648	0.3542	0.9722	1.0	0.6667	0.8125
46	41538	0.5032	0.6824	0.7632	0.4881	0.9816	0.5528	0.5189	0.5702
47	42638	0.8712	0.835	0.9318	0.8269	0.9672	0.8385	0.9233	0.8436
48	40669	1.0	1.0	0.9994	1.0	1.0	1.0	1.0	1.0
49	40681	1.0	1.0	0.9996	0.9961	1.0	1.0	1.0	1.0
50	40690	0.9999	1.0	0.9999	0.9979	1.0	1.0	1.0	0.9993
51	724	0.9229	0.9325	0.9965	0.8941	0.9965	0.9396	0.9694	0.9502
52	731	0.8201	0.6806	0.802	0.6829	0.8892	0.6875	0.9643	0.6806
53	729	0.9308	1.0	0.9962	0.9762	1.0	1.0	1.0	1.0
54	730	0.9847	0.9934	0.9998	0.9626	0.9997	0.9926	1.0	0.9778
55	726	0.8587	0.8929	0.9393	0.8052	0.9995	0.9416	0.9444	0.8279
56	767	0.9533	0.9372	0.988	0.9048	0.9699	0.9138	0.9984	0.9369
57	764	0.9157	0.9609	0.9909	0.9112	0.9816	0.9316	0.9604	0.9276
58	765	0.9783	0.931	0.9885	0.9439	0.9925	0.9569	0.9344	0.9057
59	790	0.9601	0.8958	0.994	0.9028	0.994	0.9306	1.0	0.9167
60	795	0.5192	0.5002	0.7443	0.5366	0.6927	0.4993	0.632	0.4597
61	865	0.6932	0.3542	0.6836	0.2917	0.8164	0.3542	0.9286	0.3958
62	864	0.7268	0.821	0.8446	0.8766	0.9372	0.3457	0.85	0.7037
63	867	0.8188	0.858	0.9516	0.9115	0.9442	0.8786	0.8906	0.821
64	899	0.9446	0.9077	0.9163	0.8385	0.9807	0.8692	1.0	0.8923
65	905	0.7778	1.0	0.8768	0.8571	0.9555	0.9841	0.875	0.9524
66	900	0.5454	0.516	0.8533	0.5537	0.9411	0.4286	0.6742	0.476
67	942	0.7135	0.5119	0.6364	0.4444	0.8523	0.4683	0.875	0.5476
68	944	0.8297	0.7981	0.9898	0.7691	0.8988	0.7883	0.899	0.7963
69	945	0.8858	0.8846	0.9375	0.7778	0.9979	0.8632	1.0	0.8077

	OpenML ID	Auto^{n}ML Train	Auto^{n}ML Test	H_2O Train	H_2O Test	TPOT Train	TPOT Test	AutoGluon Train	AutoGluon Test
70	946	0.6668	0.4583	0.9128	0.468	0.9324	0.5639	0.4694	0.4625
71	967	0.9763	0.9704	1.0	0.969	0.9999	0.9762	0.9748	0.9733
72	961	0.8487	0.8457	0.8514	0.8066	0.9693	0.8719	0.8489	0.8634
73	968	0.9525	0.8447	0.9495	0.8724	0.9702	0.8773	0.9758	0.8835
74	960	0.3239	0.6029	0.6867	0.4788	0.7291	0.4396	0.625	0.6324
75	996	0.9032	0.9276	0.9999	0.8947	0.999	0.8975	0.95	0.9227
76	997	0.9988	1.0	0.9985	0.9961	1.0	0.9994	0.9995	0.9984
77	1025	0.9703	0.9413	0.9885	0.9575	0.8562	0.8056	0.9836	0.9174
78	739	0.7202	0.7167	0.8919	0.6556	0.9943	0.5389	0.8	0.8167
79	733	0.9835	0.9889	0.9974	0.9947	0.9917	0.9894	1.0	0.9884
80	784	0.8509	0.8881	0.9938	0.8928	0.9749	0.8666	0.9636	0.8462
81	777	0.9065	0.9444	0.9779	0.9259	0.9773	0.8704	0.9167	0.8889
82	782	0.9861	1.0	0.998	1.0	0.9993	1.0	1.0	1.0
83	875	0.9435	1.0	0.9942	0.9854	0.9948	0.9941	1.0	0.8669
84	916	0.9113	0.7564	0.8914	0.7233	1.0	0.8354	0.9815	0.7821
85	895	0.9197	0.9473	0.9655	0.939	0.9723	0.9275	0.9451	0.9395
86	974	0.9317	0.9925	0.9579	0.9883	0.9641	0.9908	0.9394	0.94
87	754	0.9538	0.92	0.8353	0.87	0.9939	0.9444	1.0	0.8733
88	811	0.9823	0.9315	0.9964	0.9353	0.9971	0.8964	1.0	0.9522
89	747	0.9983	0.9653	0.9962	0.9769	0.9998	0.9816	1.0	0.9918
90	714	0.6244	0.377	0.6515	0.5192	0.7269	0.4584	0.7024	0.5397
91	955	0.7779	0.9	0.7088	0.7949	0.9993	0.8082	0.8917	0.7508
92	748	0.8248	0.9106	0.9975	0.8646	0.8386	0.8348	0.881	0.7545
93	719	0.8019	0.6957	0.9956	0.5749	1.0	0.6981	0.8556	0.558
94	1075	0.7375	0.8125	0.9088	0.5417	0.8935	0.2708	0.7778	0.2812
95	814	0.9262	0.9408	0.9998	0.9349	0.9995	0.9502	0.9744	0.9359
96	776	0.9377	0.8976	0.9999	0.9231	1.0	0.9526	0.9111	0.9127
97	911	0.9439	0.9628	1.0	0.9449	0.9997	0.9619	0.9972	0.951
98	886	0.6847	0.6871	0.9899	0.6272	0.9873	0.6494	0.7956	0.6812
99	796	0.9995	1.0	0.9976	0.9989	1.0	1.0	1.0	1.0
100	774	0.5631	0.5652	0.6265	0.5311	0.5243	0.4977	0.5797	0.5251
101	893	0.641	0.9167	0.6116	0.6556	0.884	0.5722	0.7667	0.8778
102	906	0.5293	0.4084	0.9256	0.4671	0.9992	0.433	0.5787	0.5392
103	884	0.9595	0.9262	0.9959	0.9373	0.9998	0.9624	0.9502	0.9654
104	894	0.9931	1.0	0.9989	1.0	0.9948	1.0	1.0	1.0
105	770	0.9995	1.0	1.0	1.0	1.0	1.0	1.0	1.0
106	870	0.9634	0.9612	0.9993	0.9438	0.9953	0.9731	0.9879	0.9449
107	749	0.9652	0.9463	0.9962	0.9764	0.9979	0.9858	0.9915	0.9799
108	1014	0.5554	0.5447	0.7281	0.4906	0.6218	0.4956	0.6293	0.4792
109	947	0.9042	0.869	0.9998	0.9373	0.9892	0.9141	0.9095	0.8142
110	841	0.9938	0.9996	0.9998	0.9987	0.9998	0.9995	0.998	0.9992
111	1005	0.8954	0.9041	0.9984	0.8602	1.0	0.9243	0.9307	0.8612
112	950	0.9252	0.9988	0.9355	0.915	0.9997	0.9946	1.0	0.9975
113	907	0.5073	0.4803	0.8244	0.5547	0.639	0.4565	0.6433	0.5821
114	874	1.0	0.9545	1.0	1.0	1.0	0.9545	1.0	1.0
115	750	0.6023	0.7786	0.7227	0.6032	1.0	0.6641	0.7037	0.6636
116	848	0.898	0.7619	1.0	0.7381	1.0	0.4444	1.0	0.6429
117	1049	0.9381	0.9435	0.9944	0.9332	0.9939	0.9381	0.9676	0.951
118	847	0.943	0.9336	0.9661	0.9363	0.9684	0.9338	0.9621	0.9387
119	316	0.8581	0.9503	0.9922	0.8662	0.9589	0.9269	0.9782	0.903
120	910	0.9734	0.981	0.9988	0.9789	0.9998	0.9819	0.9834	0.9809
121	904	0.9394	0.9653	0.9998	0.9608	0.9983	0.9558	0.9586	0.9628
122	930	0.8049	0.8502	0.9014	0.8346	0.9572	0.8553	0.8582	0.8546
123	958	1.0	0.9998	1.0	0.9982	1.0	0.9997	1.0	0.9999
124	1019	0.9998	0.9998	0.9998	0.9991	1.0	0.998	1.0	0.9997
125	723	0.9558	0.9662	0.9992	0.965	1.0	0.9778	0.9703	0.9602
126	734	0.9562	0.9559	0.9913	0.9593	0.9561	0.9529	0.9596	0.9599
127	4154	0.9586	0.8352	0.9971	0.7659	0.9996	0.8235	1.0	0.6971
128	1056	0.9387	0.8858	0.9849	0.8327	0.9985	0.8941	0.9946	0.9173
129	1002	0.8469	0.863	0.9027	0.8672	0.9141	0.8745	0.8622	0.8754
130	845	0.9522	0.9437	0.9972	0.9434	0.9994	0.9624	0.9753	0.9582
131	1020	0.9979	0.9994	0.9992	0.9981	1.0	0.9998	1.0	0.9997
132	971	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
133	44	0.9851	0.9886	0.9992	0.989	0.9978	0.9869	0.9897	0.9889
134	979	0.9599	0.9654	0.9992	0.958	1.0	0.9667	0.9758	0.965
135	718	0.961	0.9309	0.9993	0.9396	0.9979	0.9631	0.9895	0.9379
136	715	0.9707	0.9672	0.9999	0.9698	0.999	0.9758	0.9875	0.9745
137	761	0.9817	0.9821	0.996	0.9828	0.9985	0.9829	0.986	0.9854
138	1453	0.8137	0.8574	0.9871	0.7733	0.9878	0.858	0.8369	0.8397
139	821	0.948	0.9536	0.9861	0.9576	0.9339	0.9336	0.9599	0.9608

	OpenML ID	Auto^{n}ML Train	Auto^{n}ML Test	H_2O Train	H_2O Test	TPOT Train	TPOT Test	AutoGluon Train	AutoGluon Test
140	31	0.7937	0.7683	0.9925	0.7483	0.9871	0.7363	0.8413	0.7633
141	1504	1.0	1.0	1.0	0.9998	1.0	1.0	1.0	1.0
142	849	0.9511	0.9523	0.9992	0.944	0.995	0.9619	0.9657	0.9596
143	1068	0.8455	0.897	0.871	0.8038	0.9959	0.9118	0.87	0.8849
144	1022	0.999	0.9983	0.9999	0.9973	1.0	0.9998	1.0	0.9997
145	981	0.9484	0.9498	0.9891	0.9581	0.9567	0.9423	0.9637	0.9603
146	1487	0.9165	0.919	0.99	0.9348	1.0	0.9387	0.913	0.9444
147	1471	0.9789	0.988	0.9998	0.9934	1.0	0.9956	0.998	0.9981
148	995	0.9986	0.9998	0.9984	0.998	1.0	0.9994	1.0	0.9978
149	143	0.9951	0.9956	0.9966	0.9964	0.9943	0.9945	0.9972	0.9964
150	3	0.9985	0.9995	1.0	0.9998	0.9999	0.9996	1.0	0.9999
151	1496	0.9961	0.9965	1.0	0.9967	0.9978	0.9977	0.9943	0.9968
152	1461	0.9241	0.9257	0.9672	0.9326	0.9287	0.901	0.9404	0.9366
153	751	0.9629	0.9658	0.9984	0.9458	0.9965	0.9676	0.9822	0.9532
154	1067	0.8164	0.8264	0.9642	0.8085	0.9867	0.8427	0.7992	0.826
155	722	0.9985	0.999	1.0	0.9995	0.9977	0.9936	0.9994	0.9997
156	802	0.8737	0.8971	0.999	0.9143	0.9999	0.9117	0.8875	0.9228
157	1547	0.7126	0.698	0.7162	0.6886	0.8978	0.6772	0.6661	0.7274
158	913	0.9683	0.9872	1.0	0.9859	1.0	0.9888	0.9963	0.9905
159	976	0.9994	0.9998	0.9996	0.9993	1.0	0.9999	0.9997	0.9999
160	953	0.9936	0.9915	0.9999	0.9892	1.0	0.9896	0.9922	0.9908
161	993	0.993	0.9886	0.9998	0.99	0.9979	0.9897	0.9941	0.9902
162	752	0.9571	0.9566	0.9987	0.9602	0.9683	0.9496	0.9678	0.9691
163	1018	0.8942	0.9003	0.9079	0.8999	0.9496	0.8982	0.9015	0.8989
164	1050	0.8425	0.8559	0.9725	0.8293	0.977	0.8485	0.85	0.8354
165	797	0.9586	0.9657	0.9996	0.9571	1.0	0.9646	0.9779	0.9663
166	806	0.9589	0.9642	0.9999	0.9577	1.0	0.9678	0.9565	0.9563
167	866	0.9631	0.9759	0.9985	0.9663	0.9996	0.9784	0.9671	0.9657
168	837	0.9703	0.9779	1.0	0.9785	1.0	0.9819	0.984	0.9804
169	897	0.9996	0.9994	1.0	0.998	1.0	0.9996	1.0	0.9987
170	903	0.978	0.967	0.9995	0.9632	0.9998	0.9764	0.9767	0.9609
171	1494	0.9233	0.9347	0.9887	0.9172	0.9936	0.9346	0.9306	0.9309
172	917	0.9646	0.9749	0.9999	0.9782	0.9989	0.9755	0.982	0.9785
173	983	0.7576	0.7871	0.9074	0.7754	0.8441	0.7892	0.7925	0.7934
174	977	0.9998	0.9998	1.0	0.9999	0.9999	0.9998	1.0	1.0
175	1021	0.9909	0.9921	0.9966	0.9925	0.9988	0.9943	0.9942	0.9956
176	980	0.9983	0.9997	0.9999	0.9986	1.0	0.9999	1.0	0.9999

TABLE VIII

DATA TASKS DESCRIPTION

	OpenML ID	Computing Machine	No. of Features	No. of Instances
0	1013	Lab Server	3	138
1	823	Lab Server	9	20640
2	799	Lab Server	6	1000
3	1004	Lab Server	61	600
4	842	Lab Server	11	60
5	1006	Lab Server	19	148
6	737	Lab Server	7	3107
7	740	Lab Server	11	1000
8	1220	Lab Server	10	39948
9	757	Lab Server	22	528
10	792	Lab Server	6	500
11	1011	Lab Server	8	336
12	803	Lab Server	6	7129
13	13	Desktop	10	286
14	15	Desktop	10	699
15	37	Desktop	9	768
16	43	Desktop	4	306
17	50	Desktop	10	958
18	333	Desktop	7	556
19	334	Desktop	7	601
20	335	Desktop	7	554
21	346	Desktop	5	50
22	444	Desktop	4	132
23	448	Desktop	4	120
24	450	Desktop	5	264
25	451	Desktop	6	500
26	464	Desktop	3	250
27	472	Desktop	4	87
28	476	Desktop	6	50
29	479	Desktop	10	92
30	949	Desktop	5	559
31	1037	Desktop	15	4562
32	1566	Desktop	101	1212
33	744	Desktop	6	250
34	1558	Desktop	17	4521
35	1024	Desktop	35	2796
36	23499	Desktop	10	277
37	1167	Desktop	9	320
38	1511	Desktop	9	440
39	1524	Desktop	7	310
40	890	Desktop	8	108
41	1455	Desktop	7	120
42	1473	Desktop	10	100
43	1463	Desktop	6	100
44	1495	Desktop	7	250
45	40714	Desktop	6	32
46	41538	Desktop	7	246
47	42638	Desktop	8	891
48	40669	Desktop	7	160
49	40681	Desktop	7	128
50	40690	Desktop	10	512
51	724	Desktop	4	468
52	731	Desktop	5	96
53	729	Desktop	4	44
54	730	Desktop	6	250
55	726	Desktop	6	100
56	767	Desktop	4	475
57	764	Desktop	4	450
58	765	Desktop	4	475
59	790	Desktop	3	55

	OpenML ID	Computing Machine	No. of Features	No. of Instances
60	795	Desktop	4	662
61	865	Desktop	3	100
62	864	Desktop	8	60
63	867	Desktop	3	130
64	899	Desktop	6	92
65	905	Desktop	3	39
66	900	Desktop	7	400
67	942	Desktop	4	50
68	944	Desktop	10	130
69	945	Desktop	7	76
70	946	Desktop	3	88
71	967	Desktop	9	406
72	961	Desktop	8	285
73	968	Desktop	4	365
74	960	Desktop	9	90
75	996	Desktop	10	214
76	997	Desktop	5	625
77	1025	Desktop	6	400
78	739	Desktop	8	62
79	733	Desktop	7	209
80	784	Desktop	4	140
81	777	Desktop	8	47
82	782	Desktop	3	120
83	875	Desktop	4	100
84	916	Desktop	6	100
85	895	Desktop	3	222
86	974	Desktop	5	132
87	754	Desktop	6	100
88	811	Desktop	3	264
89	747	Desktop	5	167
90	714	Desktop	5	125
91	955	Desktop	6	151
92	748	Desktop	6	163
93	719	Desktop	8	137
94	1075	Desktop	9	130
95	814	Desktop	3	468
96	776	Desktop	6	250
97	911	Desktop	6	250
98	886	Desktop	8	500
99	796	Desktop	8	209
100	774	Desktop	4	662
101	893	Desktop	6	73
102	906	Desktop	8	400
103	884	Desktop	6	500
104	894	Desktop	6	66
105	770	Desktop	7	625
106	870	Desktop	6	500
107	749	Desktop	6	500
108	1014	Desktop	5	797
109	947	Desktop	5	559
110	841	Desktop	10	950
111	1005	Desktop	10	214
112	950	Desktop	5	559
113	907	Desktop	8	400
114	874	Desktop	6	50
115	750	Desktop	8	500
116	848	Desktop	6	38
117	1049	NYU	38	1458
118	847	NYU	15	6574
119	316	NYU	117	2417

	OpenML ID	Computing Machine	No. of Features	No. of Instances
120	910	NYU	11	1000
121	904	NYU	51	1000
122	930	NYU	34	1302
123	958	NYU	20	2310
124	1019	NYU	17	10992
125	723	NYU	26	1000
126	734	NYU	41	13750
127	4154	NYU	31	14240
128	1056	NYU	39	9466
129	1002	NYU	56	7485
130	845	NYU	11	1000
131	1020	NYU	65	2000
132	971	NYU	77	2000
133	44	NYU	58	4601
134	979	NYU	41	5000
135	718	NYU	101	1000
136	715	NYU	26	1000
137	761	NYU	22	8192
138	1453	NYU	38	1077
139	821	NYU	17	22784
140	31	NYU	21	1000
141	1504	NYU	34	1941
142	849	NYU	26	1000
143	1068	NYU	22	1109
144	1022	NYU	241	2000
145	981	NYU	69	10108
146	1487	NYU	73	2534
147	1471	NYU	15	14980
148	995	NYU	48	2000
149	143	NYU	17	131072
150	3	NYU	37	3196
151	1496	NYU	21	7400
152	1461	NYU	17	45211
153	751	NYU	11	1000
154	1067	NYU	22	2109
155	722	NYU	49	15000
156	802	NYU	19	1945
157	1547	NYU	21	1000
158	913	NYU	11	1000
159	976	NYU	15	9961
160	953	NYU	61	3190
161	993	NYU	61	7019
162	752	NYU	33	8192
163	1018	NYU	57	8844
164	1050	NYU	38	1563
165	797	NYU	51	1000
166	806	NYU	51	1000
167	866	NYU	51	1000
168	837	NYU	51	1000
169	897	NYU	16	1161
170	903	NYU	26	1000
171	1494	NYU	42	1055
172	917	NYU	26	1000
173	983	NYU	10	1473
174	977	NYU	17	20000
175	1021	NYU	11	5473
176	980	NYU	65	5620

ELVIS: Efficient Line Detection and Tracking for Visual Servoing

Yash Jangir[1], Junyi Geng[2] and Sebastian Scherer[3]

***Abstract*— In this work, we investigate Visual Servoing for Aerial Manipulation using a fully actuated multirotor in GPS denied and indoor environments based only on onboard processing and sensing for state estimation and servoing of multirotor. Our work focuses on developing a robust and reliable line detection and tracking approach for visual servoing systems. Our approach is designed to provide target surface edge line information in image space to controllers, which can use such information for efficient control strategies. Our approach uses a stereo camera to track wall edges and lines effectively in image space. We use depth images to calculate surface normals and effectively detect and track the target surface. We further use image segmentation to detect lines and edges only near the target surface in RGB images. Due to the surface selection, we significantly reduce the size of the RGB image to be processed, resulting in less processing time and better results. We keep track of the equation of lines in image space as the multirotor moves in 3D space and couple this system with a specialized pixel error-based control strategy that enables precise servoing. The line detection is tested on a Realsense D435 Stereo camera with Robot Operating System(ROS) for interaction tasks near a wall.**

***Index Terms*— Visual Servoing, Stereo Vision, Aerial Robotics, Depth Image**

I. INTRODUCTION

Aerial Manipulation is a growing subject of interest as it combines the capabilities of UAVs(Unmanned Aerial Vehicles) with manipulation-based robots. There are numerous advantages of using UAVs capable of physical interaction with surrounding objects. It enables UAVs to perform challenging tasks such as bridge inspection, power line inspection, painting and deburring, etc., in hard-to-reach environment areas. Much research has been conducted on developing effective control strategies for fully-actuated multirotor with manipulators; these take the aid from off-board state estimation methods, which heavily depend on the surrounding environment. In GPS-denied environments, it becomes a very challenging task to servo through the environment with high precision. Visual servoing is a very explored area of research as it provides methods to use visual sensor information for feedback to controllers for such tasks [1] This is similar to muscle-eye collaboration for organism body control. Although extracting and filtering useful visual information is a very tedious task that requires high processing power and time, this can effectively eliminate off-board information requirements from RTK GPS, motion capture system, ground station, and off-board processing or communication with the vehicle. Visual servoing particularly becomes challenging for UAV applications near walls and bridges due to continuous texture-less surfaces and fewer features for constant processing. In this paper, we have formulated an approach for such cases using minimal features near walls and bridges that are always available. We can use them to provide feedback to controllers for effective control near such surfaces.

The remainder of this paper is organized as follows: In Section II, the related work on visual servoing for different tasks and different vehicles is presented the adopted terminology and assumptions are presented. Section III refers to the Methodology of the Image Based Visual Servoing(IBVS) approach, while Section IV presents the Experimentation, and the results are discussed. Conclusions are drawn in section V, and future work is discussed in section VI.

II. RELATED WORK

In recent years there has been a lot of advancement in the application of visual servoing. It has been used in various autonomous robots and for other use cases.

Some works have used Visual servoing for landing in quadrotors. In [2], visual servoing has been used to autonomously land a UAV on a moving vehicle with the circular or elliptical pattern on the top directly in the image space, whereas [3] also uses Image-based Visual Servoing to detect and land VTOL UAV on a moving vehicle with an adaptive sliding mode controller. In [4], a deep learning-based architecture is used to detect landing sites and supply pixel coordinates in image space to servo and land. In [5], the road following for small aircraft has been achieved by tracking road lane features and using the same for landing autonomously.

Other works focus on using visual servoing for hovering tasks. [6] uses a color-based tracking algorithm to hover over a ground target. In [7], An adaptive image-based visual servo (IBVS) control for a quadrotor helicopter is proposed with a similar focus on features in image space.

Similar approaches have also been used for Visual servoing in Manipulators, [8] for example, presents an online image-based visual servoing (IBVS) controller for a 6-degrees-of-freedom (DOF) robotic system based on the robust model predictive control (RMPC) method. [8] discusses some general visual servoing-based approaches for all types of unmanned vehicles, which can be coupled with different controllers to achieve stable performance.

For Visual Servoing in aerial manipulation, [9] for instance, real-time tracking of power conductors using a

[1]Yash Jangir with Birla Institute of Technology and Science, Pilani, Goa Campus, India =f20190526@goa.bits-pilani.ac.in
[2]Junyi Geng with Robotics Institute, Carnegie Mellon University, Pittsburgh,PA junyigen@andrew.cmu.edu
[3] Sebastian Scherer with Robotics Institute, Carnegie Mellon University, Pittsburgh,PA basti@andrew.cmu.edu

Quadrotor Unmanned Air Vehicle has been achieved. The authors have proposed two approaches for image-based visual servoing solving control is directly solved in the 2D image space. They used an Image-Based Visual Servoing (IBVS) formulation combined with a Linear Quadratic Servo (LQ-Servo) in the first approach. They estimated the vehicle's partial pose concerning the lines to perform servoing in the second approach. [10] bypasses the bottleneck of time delays in signal propagation between perception and actuation modules and presents a method for estimating the perception-action time delay and its active compensation based on the predicted motion of the manipulator end-effector. [11] takes inspiration from agile, fast-moving birds for visual servoing for aerial grasping and perching. They present mathematical models and algorithms for motion planning and control, required to incorporate similar capabilities in quadrotors equipped with a monocular camera.

III. Problem Formulation

In an unknown environment, we don't effectively know how many edges, lines, and surfaces we would encounter. This creates a challenging task of selecting the significant lines or features the controller needs for effective servoing. Detecting these lines while analyzing full RGB images of high resolution creates a bottleneck in the processing and detection part, which affects the control strategy as feedback is received slowly.

In our use case, we need to servo near a wall or walls of a bridge. We formulate our situation by taking the example of a UAV near a White Board or a wall where too many features may or may not be available, but some edges are available. Fig.1 show the white board in open space.

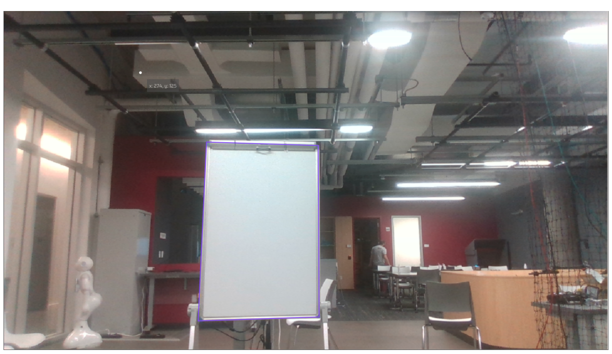


Fig. 1. This picture shows the target lines around the board in Purple

IV. Methodology

Our approach consists of leveraging the depth image to calculate surface normals. We find the surface normals using the depth gradient method and then apply Median filter and Bilateral filter to perform hole filling in the normal image. We also use a temporal filter on normal images which is a sliding window which performs averaging over the window.

We initialize a surface and find boundaries of the target surface as a bounding box. We assume that we can initialize one point on the surface that we are tracking in the camera frame as pixel value. We use simple 8 direction search around the initial surface point selected to find surface boundary. We keep updating the new initial point as the centeroid of old bounding box and track the surface using that.

Further we use this bounding box to segment the surface part in the RGB image. On this smaller image we use a bilateral filter, canny edge detection and probabilistic Hough Transform algorithm to detect lines only near the target surface. This approach decreases our computation and helps us use depth information as a filter for rejecting lines and edges, which are present in the open scene. This is further passed to a Final Processing stage which uses temporal filtering, low pass filtering and averaging of lines on basis of slope to get stable lines. Fig.2 shows a high level block diagram of the system. Each step has been described in detail further below.

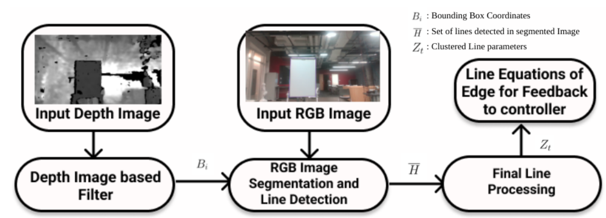


Fig. 2. This picture shows the target lines around the board in Purple

A. Depth Image Filtering and Surface Normal Generation

The raw depth image received has several different noises. Firstly, the depth image has a high salt and pepper noise due to continuously changing pixel values. Secondly, there are a lot of points for which the depth information is not received, and hence they create holes in the depth image. Further, overall smoothing of the depth image is also required. For this purpose, we pass the depth image with simple filters to get a better image.

We pass the image through a Median Filter [12] to reduce salt and pepper noise in the depth image. This filter replaces each pixel value with the median of neighboring pixels. After that, The depth image is passed through a Gaussian filter [13] to further smooth out the depth image. Thresholding is performed on the depth image to remove disturbances from far away and too close points.

We calculate the surface normal vectors using depth data. Consider the camera frame's 3D points to be (X, Y, Z). We define (u,v) in the image plane given by Equation 1, where Z is the pixel depth, f_x and f_y are focal lengths of the camera, and C_x and C_y are cameras intrinsic. We calculate the normal vector using equations 1-4, where $\vec{n}$ is the normal vector.

$$\begin{aligned} u &= \frac{x}{z} \cdot f_x + c_x \\ v &= \frac{y}{z} \cdot f_y + c_y \end{aligned} \tag{1}$$

$$
\begin{aligned}
\vec{n}' &= \left(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, 1\right) \\
\vec{n} &= \frac{\vec{n}'}{\|\vec{n}'\|}
\end{aligned} \tag{2}
$$

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial u} \cdot \frac{f_x}{z} \approx \frac{\Delta z}{\Delta u} \cdot \frac{f_x}{z} \tag{3}$$

$$\frac{\partial z}{\partial y} = \frac{\partial z}{\partial v} \frac{f_y}{z} \approx \frac{\Delta z}{\Delta v} \cdot \frac{f_y}{z} \tag{4}$$

After finding the surface normals, we add each normal image to a buffer of a predefined temporal sliding window we also replace all the holes in the image by zeros. We further then average the sliding window to perform image hole filling. Assigning holes to zeros instead of normal of infinite depth surface stops the holes from effecting the average. Once we complete the averaging we re-assign all the holes back to infinite depth surface normal value. We pass the normals to the surface detection and tracking module.

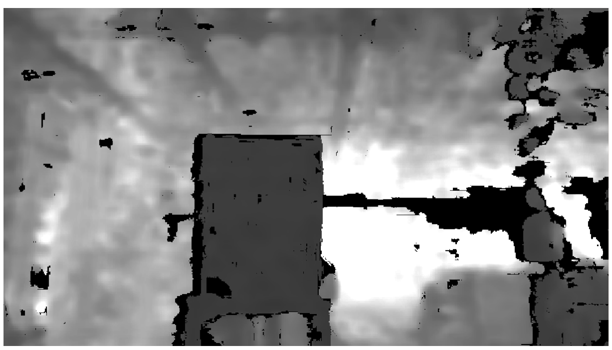


Fig. 3. Aligned Depth image to RGB image by Realsense D435

B. Surface detection and Tracking

We assume that the target surface is initially at the center of the camera frame, which helps initialize our starting point of the search on the target surface. We then scan in eight directions in a straight line around the point until we find a different normal vector in each direction in the image space to find eight boundary coordinates around the surface, comparing the average value of surface normal on each step with the last step. We know that the centroid of these points will lie on the target surface, assuming that camera displacement is not very high. So, we update the new search point for the following image as the centroid of the boundary points. We then take the bounding box using these points around the surface. This strategy helps in continuous detection and tracking of the surface. Fig.4 shows the bounding box generated in normal image using our algorithm.

C. Line Detection

In this step, we use a depth-aligned RGB image to detect lines. We first use the bounding box detected from the Surface detection and Tracking algorithm to segment the

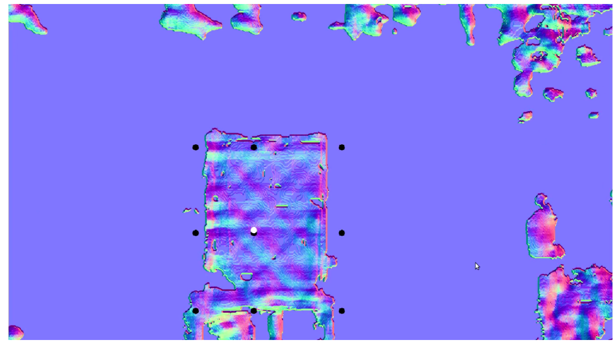


Fig. 4. Surface scanning and tracking with bounding box

target surface part of an image. We use a suitable extra border strip around the bounding box for better edge detection. The Surface RGB image is passed from a canny edge detector [14] which gives a binary image. This image is then passed with Probabilistic Hough Transform [15]. Hough Transform detects straight lines. The lines detected in the image are then used for final processing.

D. Final Processing

We first find out the equation of edge lines of the wall, which are visible in image space. These can be denoted as :

$$\lambda_i = 0 \tag{5}$$

where $i \in [0, n]$ where n is number of lines. Further, each line equation can be denoted in the standard form of line as equation 6.

$$\alpha X + \beta Y + \gamma = 0 \tag{6}$$

Where α, β, and γ are line parameters to be found out for each line while X and Y are pixel points in image space lying on the line.

We target to get any 2 points$[(x_1, y_1), (x_2, y_2)]$ on this line to find α, β using two point form of line. We group the lines on basis of slope and intercept. We then average the line groups. This set of average lines is provided as feedback and plotted on the RGB image. This gives us the significant edges of the surface accurately.

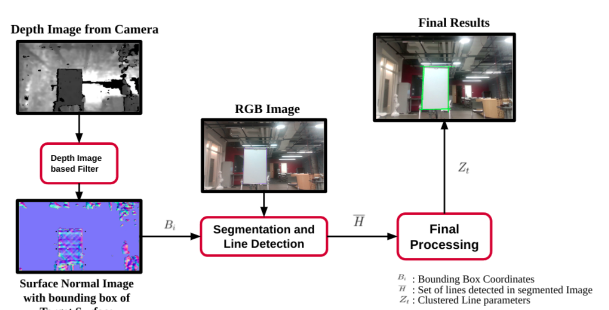


Fig. 5. Overview of the System

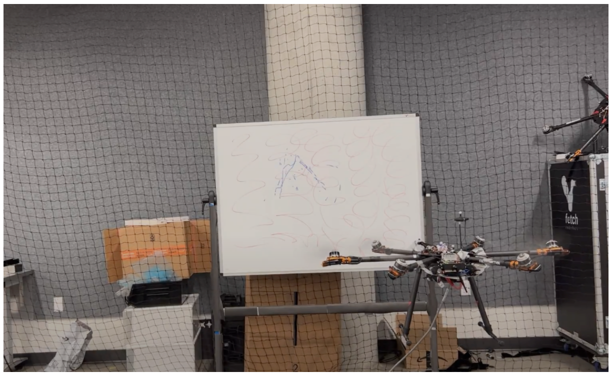


Fig. 6. Testing with Fully Actuated UAV on right edge of surface

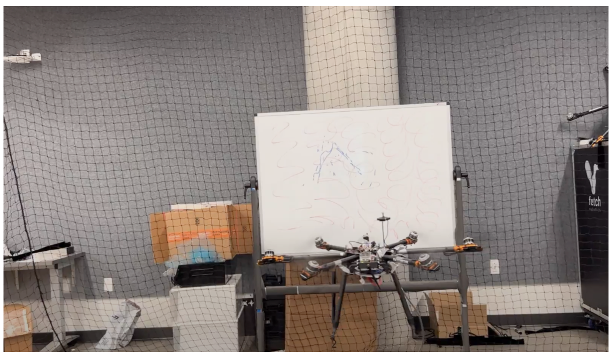


Fig. 7. Testing with Fully Actuated UAV on central bottom of surface

V. Implementation and Experimentation

We have used a Real-sense D435 stereo camera with ROS(Robot Operating System) for experimentation. The python code is run on an Intel i7 9th Gen 3.6Ghz CPU with about 16 GB RAM. No discrete GPU is utilized to process the images. We use the ROS package for the real sense to get aligned depth to the RGB image. We have used the OpenCV2 [16] library for implementing Gaussian Filters, Median Filters, Canny Edge Detection, and Probabilistic Hough Transform. We use the same library for image transformation segmentation and format conversion. The whole implementation is of algorithm and surface normal calculations have been written in Python3 using Scipy [17] and Numpy [18] libraries. We directly experiment using the real sense rather than using a simulation environment.

VI. Conclusion and Results

We tested our algorithm in open scene on target surface Fig. 8 show the all lines without our system and Fig.9 show the results of our algorithm. Fig.5 shows step wise results of the system. We tested our implementation with Realsense D435 mounted on a fully actuated UAV. Our algorithm working in air to provide visual feedback for the controller. Fig.6 and Fig.7 show the UAV moving while Fig.10 and Fig.11 show the results of our algorithm in flight. The final rate of line parameter supply is 23Hz while visualizing the lines on RGB. The rate of feedback without visualization libraries is 24Hz.

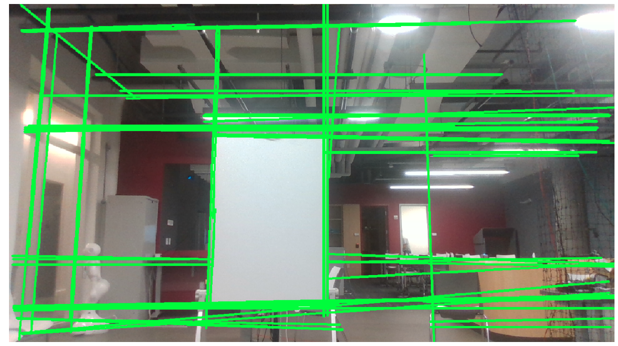


Fig. 8. Raw Lines

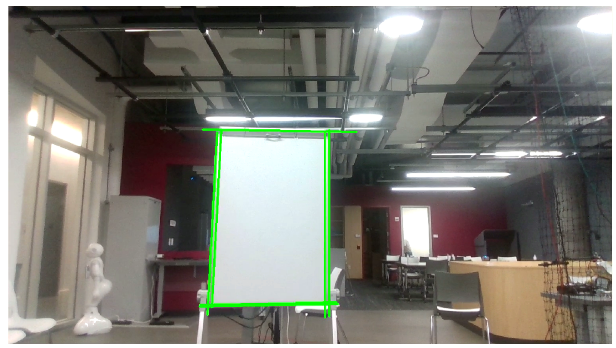


Fig. 9. Final Results Line Detection with our approach

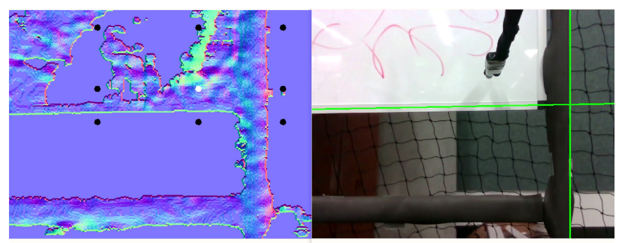


Fig. 10. Realtime Line detection on UAV at Servoing right edge of surface

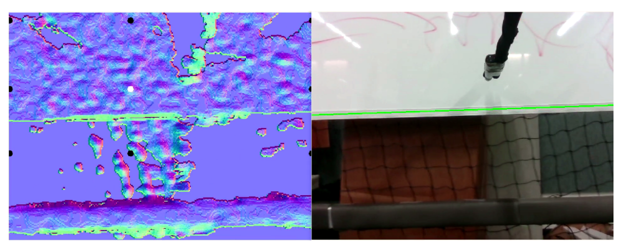


Fig. 11. Realtime Line detection on UAV at central bottom of surface

We complete the objective of line detection with high accuracy and a very high rate. Our approach provided a simple yet elegant line detection and tracking solution for Image-based visual servoing.

VII. Future Work

The depth image received is unstable, with much filtering required. We want to make the system more reliable by incorporating decimation and spatial and temporal filtering. We also plan to add a separate pipeline that can also pin parallel can use all available features by using Oriented FAST and Rotated BRIEF(ORB) with Fast Library for Approximate Nearest Neighbors (FLANN) for feature detection and matching with a Random sample consensus(RANSAC) which will also help in tracking and filtering outliers and also give camera displacement in an image frame.

References

[1] G. F. Silveira, J. R. Azinheira, P. Rives, and S. S. Bueno, "Line following visual servoing for aerial robots combined with complementary sensors," in *ICAR*, vol. 3, 2003, pp. 1160–1165.

[2] A. Keipour, G. A. Pereira, and S. Scherer, "Real-time ellipse detection for robotics applications," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7009–7016, Oct 2021.

[3] D. Lee, T. Ryan, and H. J. Kim, "Autonomous landing of a vtol uav on a moving platform using image-based visual servoing," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 971–976.

[4] P. Mathur, Y. Jangir, and N. Goveas, "A generalized kalman filter augmented deep-learning based approach for autonomous landing in mavs," in *2021 International Symposium of Asian Control Association on Intelligent Robotics and Industrial Automation (IRIA)*, 2021, pp. 1–6.

[5] E. Frew, T. McGee, Z. Kim, X. Xiao, S. Jackson, M. Morimoto, S. Rathinam, J. Padial, and R. Sengupta, "Vision-based road-following using a small autonomous aircraft," in *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)*, vol. 5, 2004, pp. 3006–3015 Vol.5.

[6] S. Azrad, F. Kendoul, and K. Nonami, "Visual servoing of quadrotor micro-air vehicle using color-based tracking algorithm," *Journal of System Design and Dynamics*, vol. 4, no. 2, pp. 255–268, 2010.

[7] D. Lee and H. J. Kim, "Adaptive visual servo control for a quadrotor helicopter," in *ICCAS 2010*. IEEE, 2010, pp. 1049–1052.

[8] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[9] O. Araar and N. Aouf, "Visual servoing of a quadrotor uav for autonomous power lines inspection," in *22nd Mediterranean Conference on Control and Automation*, 2014, pp. 1418–1424.

[10] M. Laiacker, F. Huber, and K. Kondak, "High accuracy visual servoing for aerial manipulation using a 7 degrees of freedom industrial manipulator," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1631–1636.

[11] ——, "High accuracy visual servoing for aerial manipulation using a 7 degrees of freedom industrial manipulator," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1631–1636.

[12] L. Hayat, M. Fleury, and A. Clark, "Two-dimensional median filter algorithm for parallel reconfigurable computers," *Vision, Image and Signal Processing, IEE Proceedings -*, vol. 142, pp. 345 – 350, 01 1996.

[13] R. Haddad and A. Akansu, "A class of fast gaussian binomial filters for speech and image processing," *IEEE Transactions on Signal Processing*, vol. 39, no. 3, pp. 723–727, 1991.

[14] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[15] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[16] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[17] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[18] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sept. 2020. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2

[19] J. Wu, *Advances in K-means clustering: a data mining thinking*. Springer Science & Business Media, 2012.

Model Pruning for Efficient Object Tracking

Yucong Chen[1], Yaoyu Hu[2] and Sebastian Scherer[2]

***Abstract*— There is a trend of utilizing deep learning algorithms on autonomous aircraft for situational awareness and decision-making. However, the performance of these deep learning models is restricted due to the limited onboard computation constraint and the real-time requirement. We take the visual-based object tracking model as an example and apply model pruning to reduce model complexity for better efficiency. While deep learning models are often thought to benefit from their large number of computational parameters, we find that many of them are non-essential but consume a lot of computing resource and time. The pruned model has significantly reduced number of parameters, making it capable of being deployed on an aircraft for efficient real-time inference. At the same time, the light version of the model still has comparable performance to the original one. Model pruning generally improves the usability of deep learning algorithms on autonomous aircraft.**

***Index Terms*— Object Detection, Segmentation and Categorization; Model Pruning**

I. INTRODUCTION

Object tracking is one of the fundamental tasks for autonomous aircraft. In some scenarios, the aircraft needs to identify and track other flying objects it encounters and reroute its path to avoid collisions [1]. With the development of the object detection methods [2]–[4], people start to utilize the powerful object detection methods and then apply tracking algorithms to the detected objects, which is called tracking-by-detection [5]–[7].

Early tracking algorithms without detection were simple and fast beacuse they focused on the interest points only. Despite the significant improvement in accuracy, the detection-based tracking algorithms using deep neural networks cost much more power and time. The heavy computational cost leads to difficulties of deploying the model on mobile computers on board [8].

On the other side, deep neural networks are often thought to be over-parameterized. There are some parameters of a large-scale model that are redundant to our goal [9]. However, a larger model still benefits from the higher probability of getting a set of well-learned parameters. Due to the imperfection of the initialization, if we use a small model to remit the computational cost in the first place, the accuracy will generally degenerate.

Model pruning, on the other hand, has been acknowledged as a good compromise to it. Due to the good parameters initialization inherited from the original model, the pruned model can easily reach the comparable performance of the original one, and at the same time, has a small computational cost [9].

[1] ShanghaiTech University, Shanghai, China
[2] Carnegie Mellon University, Pittsburgh, PA, USA

To this end, we introduced model pruning techniques into the workflow of object tracking by pruning the original model pretrained on the Airborne Object Tracking (AOT) dataset [10]. After the pruning, we conducted fine-tuning and performance evaluation on our self-collected dataset. The results demonstrate that model pruning can significantly improve the usability of object tracking algorithms on computation-constrained robot platforms.

II. RELATED WORK

A. Efficient Object Detection and Tracking

Object detection algorithms represented by YOLO series [2], [11]–[13], Faster R-CNN [14] and attention-based methods like [15] have been playing a remarkable role in many fields. Moreover, recent work proposed frameworks that do not require designing a set of anchor boxes [3], [4], [7], as our model takes this approach.

The application and usability of object tracking on flying drones has also been explored in previous researches [16], [17].

B. Model Compression via Pruning

The need of model pruning comes from two aspects: overfitting and high computational cost of over-parameterized models. For example, Denil et al. [18] showed how a network might be effectively rebuilt using just a tiny subset of its initial parameters. However, since we do not know how to initialize the subset of the parameters, model pruning is here to find a way to select this subset after the primary training. Researchers have verified the effectiveness of model pruning [9] and have proposed a series of pruning techniques [19]–[22]. Other methods to imporve the efficiency include knowledge distillation [23] and neural architecture search [24]–[26], which are more complex and require additional prior information compared to model pruning. As a result, we use model pruning as the compression method in our model.

III. METHODOLOGY

The overall workflow can be divided into three parts: train the original model, perform pruning algorithms and fine-tune the model until we get satisfactory performance. The following is the detailed information about the tracking framework and the pruning techniques we used.

A. Model Overview

The model we used for object tracking is illustrated in Fig. 1. The model consists of a motion estimation stage, a detection and tracking stage and some post-processing

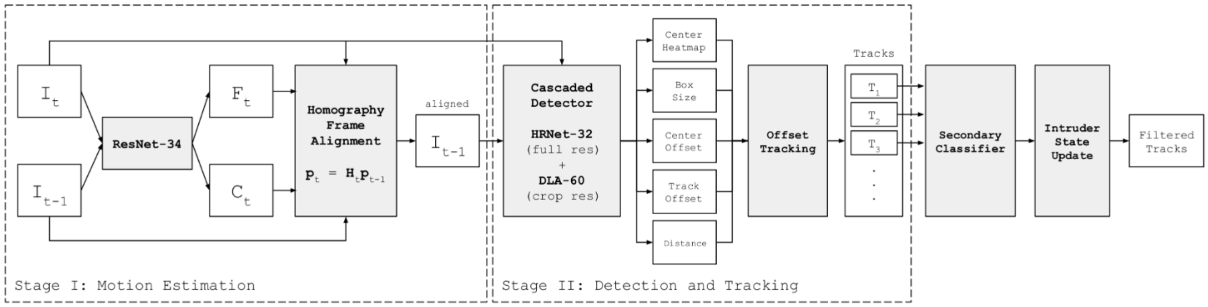


Fig. 1. Model Overview

modules. The input is two frames of a video stream, and the first stage estimates the motion of camera between the two frames and the second stage is the primary part to do object detection and tracking. The tracked items are then passed to the post-processing modules to generate the final results, which can guide the aircraft to avoid collisions and re-route its path.

TABLE I
TIME PROFILE OF THE MODEL

Stages	Time per frame
Motion Estimation	$\sim$ 0.03s
Detection & Tracking	$\sim$ 0.12s
Post-processing	$\sim$ 0.00s

The time profile of the model is shown in TABLE I generated by the inference time on AOT dataset. We can see that the Detection & Tracking stage is the most time-consuming part of the model, which means that it is the bottleneck worth to optimize. The main part of this stage is the detection model. The architecture of it is derived from the HRNet [27], [28] and we use the HRNet-32 that has 30 million parameters. Therefore, we apply model pruning to the detection model to maximize the gain from it.

B. Pruning Techniques

We used Neural Network Intelligence (NNI) [29] as the toolbox to perform model pruning.

The pruning method we take is to select a subset of the convolutional filters inside each layer. The selection precedure can be performed by many strategies. Here we take the strategy that uses the L1-norm of the filter weights as a metric to select them:

$$\mathbf{x}^i = \begin{bmatrix} x_1^i \\ x_2^i \\ \vdots \\ x_n^i \end{bmatrix} \tag{1}$$

$$\|\mathbf{x}^i\|_1 = \sum_{j=1}^{n} |x_j^i| \tag{2}$$

Where $\mathbf{x}^i$ is the flattened tensor of the ith filter. The basic assumption of using norm as the metric is that, filters with larger weights (and so larger norms) are more important than others. The 'more important' means that they have larger impacts to the model so we may prefer to keep them.

Fig. 2 demonstrates the pruning workflow. For each layer, we have different weights for each filter, here we use single numbers to represent the weights (instead of showing the 4-D tensors). The pruner will then take the weights as input, and then choose the pre-defined number of filters with the largest L1-norms of their weights. Specifically, the way that pruner selects the filters is generating a mask and then do multiplication with the filter weights. After that, some of the filters will become zero vectors and then the 'speed up' module will remove them to get the smaller set of filters.

Once we finish pruning the model, the number of parameters will be reduced so that we have a lower computational cost. Now that we are using a subset of the original model parameters to do the inference, we need to fine-tune the pruned model (i.e. do a short term training) to force the pruned model to adapt to the dataset, making it compensate for the negative effects from pruning.

IV. EXPERIMENTS

A. Datasets

We used two datasets for our experiments: the Airborne Object Tracking (AOT) dataset [10] and our self-collected sequences on flying aircraft. A sample visualization of the both datasets is shown in Fig. 3.

1) AOT Dataset: The AOT dataset is a collection of flight sequences captured by high-resolution cameras mounted on aerial vehicles, proposed in the challenge [30]. Two aircrafts fly prearranged encounters while being sensor-equipped in order to produce certain sequences. Each frame is an image of 2448×2048 resolution. We used this dataset to train the original model.

2) Self-collected Dataset: This dataset contrains eighteen sequences of flying aircrafts. The videos are captured by our own flying aircrafts and its resolution is 2432×2048. The dataset is much smaller than the AOT dataset. We used splits of this dataset to fine-tune the pruned model and do the evaluation.

B. Experiment Details

We set up five experiments under different pruning settings. They are the original model, the models with the

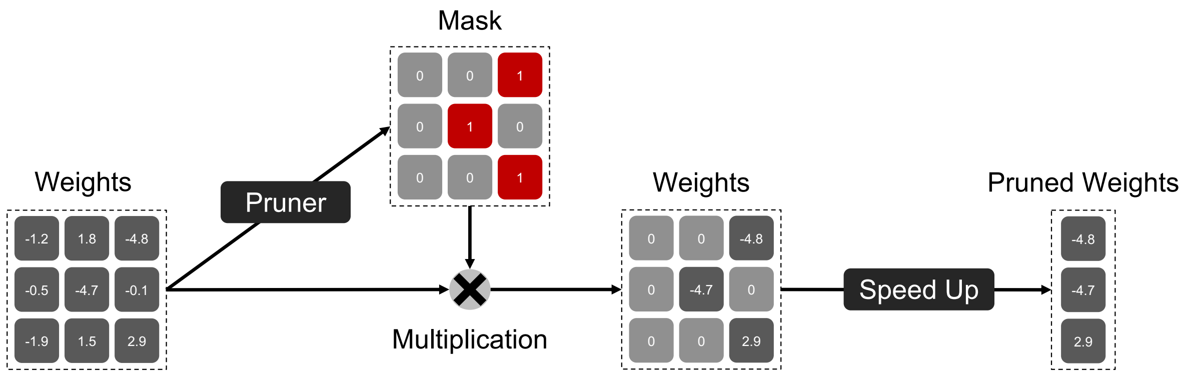
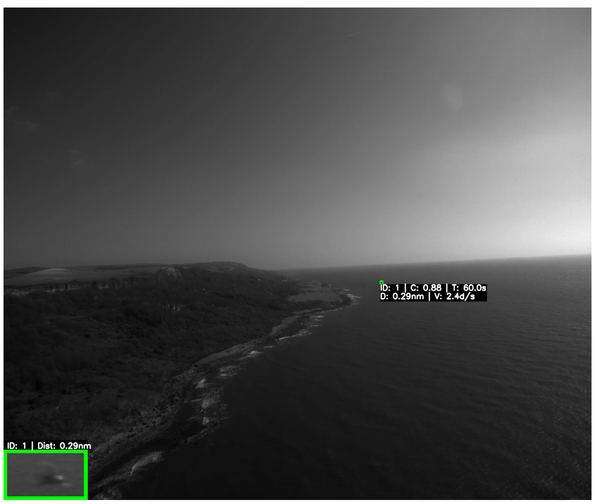


Fig. 2. Pruning Workflow



(a) AOT dataset

(b) Self-collected dataset

Fig. 3. Frames with detected item

sparsity of 0.25, 0.50, 0.75 and 0.90. The sparsity represents the intensity of the pruning. For example, if the sparsity is 0.25, then we will remove 25% of the filters in each layer.

Firstly the original model was trained on the entire AOT dataset. This training procedure was performed on one NVIDIA V100-32 GPU and it took about one week to finish. We set the bath size to 16 and we resized the input to quarter resolution due to the memory limitation.

The four pruned models were then derived from the original model. We performed the fine-tuning to all of the five models on the self-collected dataset for 15 epochs. The validation curves were successfully converged during the fine-tuning. We split the 18 sequences to be training, validation and test datasets with the proportion of $13 : 2 : 3$, we used the validation set to make sure that our fine-tuning procedure will not make the models overfitting. The fine-tuning procedure was performed on four NVIDIA V100-32 GPU in parallel with full resolution input.

C. Evaluation Metrics

As we focusing on the detection task, we used the mean IOU [31], Precision, Recall and F1-score as the performance metrics. In our self-collected data, We make sure that there is at most one object in each frame.

$$\text{mean } IOU = \frac{1}{n}\sum_i \frac{\text{Intersection}(GT_i, \text{output}_i)}{\text{Union}(GT_i, \text{output}_i)} \tag{3}$$

where n is the number of frames in the sequence.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

$$\text{F1-score} = \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

The way to define the trune positive is that the IOU between the bounding boxes of ground truth and the output is larger than 0.5.

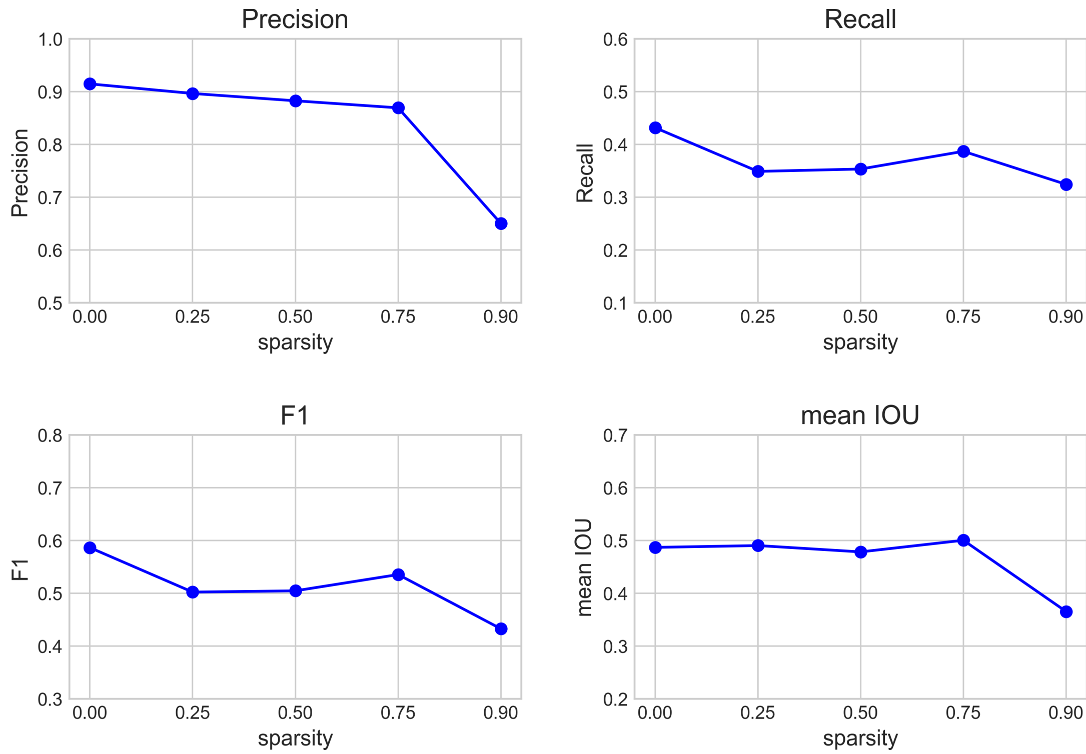


Fig. 4. Performance of each pruned model and the original model

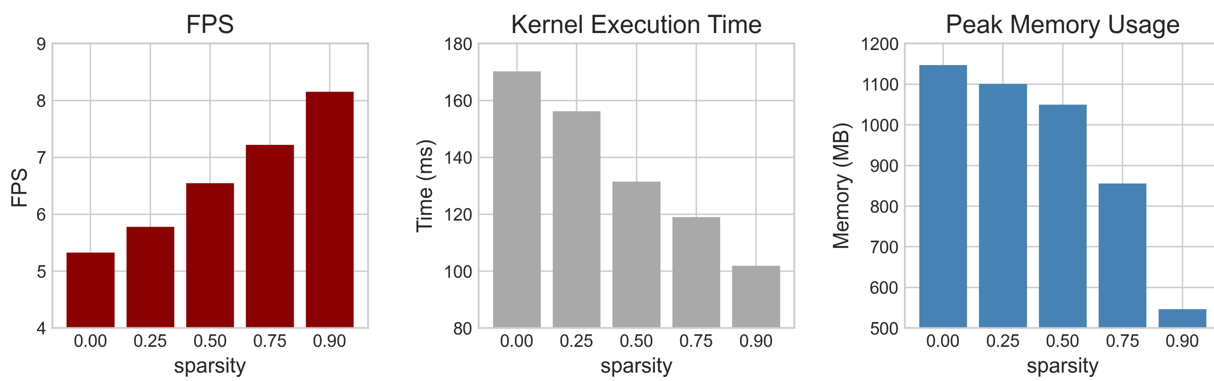


Fig. 5. Computational cost of each pruned model and the original model

As for the consumption metrics, we profile the inference FPS, Kernel Execution Time (the actual computation time on GPU) and Peak Memory Usage of the model.

D. Results

The results of the performance can be found in Fig. 4.

As we can see, the precision goes down as the sparsity increases, which means that the model becomes less capable of detecting the objects. The precision drops rapidly when the sparsity goes from 0.75 to 0.90. The recall and F1-score also drop as the sparsity increases, but we can find that when the sparsity is 0.75, they become better. The mean IOU even shows that the model with the sparsity of 0.75 is better than the original model (i.e. the sparsity of 0.00). This phenomenon may come from the decrease of the overfitting.

These four curves overall indicate that the performance of the pruned model stays stable as the sparsity located in the range of 0.00 to 0.75.

The consumption of each can be seen in Fig. 5. It is clear that the inference speed goes up as we increase the pruning intensity. Also, we can see that the memory usage has been significantly dropped after the pruning.

V. DISCUSSIONS

With the above results, we can find that 0.75 is a turning point for the model performance, and there is almost no performance decrease from the original model to the model with a sparsity of 0.75. The results suggest that the majority of the parameters of our original model are not essential for the goal. We can even remove $3/4$ of the parameters safely and still achieve the comparable performance. At the same time, the increase of the sparsity directly leads to the improvement of inference efficiency so that we can benefit from both time and memory usage.

Under this settings, we can choose the pruned model with the sparsity of 0.75 as the one that balances the performance and the efficiency best.

To this end, we have shown that model pruning will be able to increase the potential of the tracking model to be deployed on board.

ACKNOWLEDGMENT

The authors would like to thank the Robotics Institute Summer Scholars program for providing the opportunity to participate in the research. We would also give our special thanks to Dr. John M. Dolan and Ms. Rachel Burcin for bringing this amazing experience to us.

REFERENCES

[1] J. How, C. Fraser, K. Kulling, L. Bertuccelli, O. Toupet, L. Brunet, and N. Roy, "Increasing autonomy of UAVs: Decentralized CSAT mission management algorithm," *IEEE Robotics and Automation Magazine*, vol. 16, no. 2, pp. 43–51, 2009.

[2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[3] H. Law and J. Deng, "CornerNet: Detecting Objects as Paired Keypoints," *International Journal of Computer Vision*, vol. 128, no. 3, pp. 642–656, Mar. 2020.

[4] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as Points," Apr. 2019.

[5] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3464–3468.

[6] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3539–3548.

[7] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking Objects as Points," in *Computer Vision – ECCV 2020*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 474–490.

[8] M. Lofqvist and J. Cano, "Accelerating Deep Learning Applications in Space," July 2020.

[9] M. Zhu and S. Gupta, "To prune, or not to prune: Exploring the efficacy of pruning for model compression," Nov. 2017.

[10] Amazon, "Airborne Object Tracking Dataset," https://registry.opendata.aws/airborne-object-tracking/.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[12] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[13] ——, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[15] T. Meinhardt, A. Kirillov, L. Leal-Taixé, and C. Feichtenhofer, "TrackFormer: Multi-Object Tracking With Transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8844–8854.

[16] R. Barták and A. Vykovský, "Any Object Tracking and Following by a Flying Drone," in *2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI)*, Oct. 2015, pp. 35–41.

[17] P. Chen, Y. Dang, R. Liang, W. Zhu, and X. He, "Real-Time Object Tracking on a Drone With Multi-Inertial Sensing Data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 131–139, Jan. 2018.

[18] M. Denil, B. Shakibi, L. Dinh, M. Ranzato, and N. De Freitas, "Predicting parameters in deep learning," *Advances in neural information processing systems*, vol. 26, 2013.

[19] Q. Huang, K. Zhou, S. You, and U. Neumann, "Learning to prune filters in convolutional neural networks," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 709–718.

[20] Y. He, X. Zhang, and J. Sun, "Channel Pruning for Accelerating Very Deep Neural Networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1389–1397.

[21] J.-H. Luo, H. Zhang, H.-Y. Zhou, C.-W. Xie, J. Wu, and W. Lin, "ThiNet: Pruning CNN Filters for a Thinner Net," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 10, pp. 2525–2538, Oct. 2019.

[22] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," *arXiv preprint arXiv:1608.08710*, 2016.

[23] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," Mar. 2015.

[24] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015.

[25] H. Peng, H. Du, H. Yu, Q. Li, J. Liao, and J. Fu, "Cream of the crop: Distilling prioritized paths for one-shot neural architecture search," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[26] Y. Zhang, Z. Lin, J. Jiang, Q. Zhang, Y. Wang, H. Xue, C. Zhang, and Y. Yang, "Deeper insights into weight sharing in neural architecture search," *arXiv preprint arXiv:2001.01431*, 2020.

[27] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang, "High-resolution representations for labeling pixels and regions," *arXiv preprint arXiv:1904.04514*, 2019.

[28] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *TPAMI*, 2019.

[29] Microsoft, "Neural network intelligence," Jan. 2021.

[30] "AIcrowd — Airborne Object Tracking Challenge — Challenges," https://www.aicrowd.com/challenges/airborne-object-tracking-challenge.

[31] P. Jaccard, "The distribution of the flora in the alpine zone.1," *New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912. [Online]. Available: https://nph.onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8137.1912.tb05611.x

Primal-Dual Augmented Lagrangian Solver for Model Predictive Control

Zhengyu Fu[1], Kevin Tracy[2], and Zachary Manchester[2]

***Abstract*— Recently, augmented Lagrangian (AL) has been increasingly used as a model predictive control (MPC) solver to handle constrained problems speedily. With sufficiently large penalties, augmented Lagrangian exhibits a fast convergence rate theoretically. However, in practice, the penalized problem becomes increasingly ill-conditioned as the penalties are increased, making the sub-optimization problems more and more difficult to solve at each iteration. This paper proposes a novel primal-dual formulation of augmented Lagrangian that can greatly mitigate the aforementioned numerical issue and speed up the convergence. Instead of minimizing the primal augmented Lagrangian directly, the proposed optimizes a strategically constructed dual problem. We show that the contracted dual problem is equivalent to the original problem but has a significantly better numerical condition. With such a formulation, primal-dual augmented Lagrangian (PDAL) is compatible with a more aggressive penalty update and converges to the optimum more quickly. We compare the proposed method against existing MPC solvers over a trajectory optimization problem of a 4D double integrator.**

***Index Terms*— Optimization and Optimal Control**

I. INTRODUCTION

Model predictive control (MPC) provides a systematic approach to control complex under-actuated robotic systems with state and input constraints [1]. The main step of every MPC iteration is to solve a finite horizon, discrete optimal-control problem (OCP) in the following form

$$\begin{array}{ll} \underset{z}{\text{minimize}} & \frac{1}{2} z^T H z + h^T z \\ \text{subject to} & Gz = g, \\ & Cz \leq c, \end{array} \tag{1}$$

where $z \in \mathbb{R}^n$ is the decision variable; $H \in \mathbb{S}_+^n$ is a positive semi-definite cost matrix; a vector $h \in \mathbb{R}^n$; a linear equality matrix $G \in \mathbb{R}^{p \times n}$ and a linear inequality matrix $C \in \mathbb{R}^{q \times n}$.

The solution methods for (1) have been studied for many years and several methods have been proposed [2]–[4]. For example, the interior-point method approximates the inequality constraints with parameterized barrier functions, and solves a sequence of equality constraints with Newton's method; see [5] for details. However, when the objective function is changed, the optimal solution of the previous problem is far from the central path of the new problem [6], making it hard to warm start and cannot benefit from the similarities of consecutive MPC iterations.

Unlike the interior-point method which requires all intermediate solutions to be strictly feasible, penalty methods penalize constraint violation by absorbing constraints into the objective function [7]. To better approximate the original constrained problem, one common approach is to gradually increase penalty weights until reaching a satisfactory constraint tolerance. However, penalty methods usually suffer from the well-known numerical ill-conditioning issue [8] and will not be able to converge to the optimum before being ill-conditioned.

The primal augmented Lagrangian approach tries to overcome the aforementioned ill-conditioning problem by augmenting an extra term to the objective to estimate the Lagrangian multiplier associated with the constraint [7]. An unconstrained problem with fixed penalty weights and dual variables is solved in the inner loop and penalty weights are explicitly off-loaded to the augmented multiplier during the dual update in the outer loop. Though penalty weights do not have to grow unbounded before a solution within a given tolerance is found, the condition number of the KKT matrix associated with the unconstrained problem will still grow as the penalty is kept increasing for better constraint satisfaction.

Recently, first-order methods have received increasing attention in MPC. Such methods use only the first-order information to compute the optimal solution iteratively. For example, the alternating direction method of multipliers (ADMM) [4] first splits problems into smaller pieces and updates primal variables by optimizing the augmented Lagrangian. The dual variables are updated by dual ascent. Compared to second-order methods, first-order methods are generally cheap to compute per iteration and have inherent numerical stability due to the absence of the Hessian. However, their convergence rate is typically sub-linear compared to the linear convergence rate of augmented Lagrangian.

To address the ill-conditioning issue that remains in the primal augmented Lagrangian, we propose a primal-dual augmented Lagrangian formulation by introducing several additional dual variables. In contrast to the primal method, the proposed method has better numerical properties and does not impose any requirement on the rank of the gradient of the constraints.

In the remainder of the paper: Section II reviews the background of the optimality condition and the augmented Lagrangian. Section III presets the proposed method in details. Comparisons between the proposed method and other state-of-art MPC solvers are shown in Section IV. Finally, a discussion of the future directions is summarised in Section

[1]Zhengyu Fu is with the Division of Integrative Systems and Design, Hong Kong University of Science and Technology, Hong Kong, China. zfuaj@connect.ust.hk

[2]Kevin Tracy and Zachary Manchester are with The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA {ktracy,zacm}@cmu.edu

V.

II. BACKGROUND

A. Optimality conditions

The Lagrangian of (1) is usually written as:

$$\begin{aligned}\mathcal{L}(z,\lambda,\mu) =& \frac{1}{2}z^T Hz + h^T z \\ &+ \lambda^T(Gz-g) + \mu^T(Cz-c).\end{aligned} \tag{2}$$

And the first-order necessary conditions for optimality, also known as the KKT conditions, are as the following:

$$Hz + h + G^T\lambda + C^T\mu = 0 \tag{3}$$
$$Gz - g = 0 \tag{4}$$
$$Cz - c \leq 0 \tag{5}$$
$$\mu \geq 0 \tag{6}$$
$$(Cz-c)^T\mu = 0 \tag{7}$$

Particularly, (3) is referred to as the stationary condition and (7) is known as the complementary condition. If a set of primal and dual variables satisfy the KKT conditions, a local optimal solution is found.

B. Augmented Lagrangian method

Suppose to solve (1) with the augmented Lagrangian method, a wide range of penalty functions can be used to penalize the constraint violation. In this paper, we focus on the quadratic penalty which is natural and also widely used. By adding two sets of penalties with a penalty weight $\rho \in \mathbb{R}_+$, the augmented Lagrangian is defined as:

$$\begin{aligned}\mathcal{L}_\rho(z,\lambda,\mu,\rho) =& \frac{1}{2}z^T Hz + h^T z \\ &+ \lambda^T(Gz-g) + \frac{\rho}{2}(Gz-g)^T(Gz-g) \\ &+ \mu^T(Cz-c) + \frac{\rho}{2}(Cz-c)^T I_\mu (Cz-c)\end{aligned} \tag{8}$$

where $I\mu$ is a diagonal selection matrix zeroing out the inactive constraints by the following logic:

$$I_\mu(i,i) = \begin{cases} 1 & (\mu_i > 0) \vee ([Cz-c]_i > 0) \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Augmented Lagrangian follows a two-level nested loop procedure to solve the problem. It first minimizes the augmented Lagrangian with respect to the primal variable z in the inner loop and then updates the dual variables and the penalty in the outer loop. An outline of this algorithm is listed below:

1) Minimize the augmented Lagrangian with respect to z while keeping the dual variables and penalty constant:
$$z^* = \min_z \mathcal{L}_\rho(z,\lambda,\mu) \tag{10}$$
2) Update the dual variables:
$$\lambda = \lambda + \rho(Gz^* - g) \tag{11}$$
$$\mu = \max(0, \mu + \rho(Cz^* - c)) \tag{12}$$
3) Update penalty:
$$\rho = \rho\phi \tag{13}$$

When minimizing (10), the primal augmented Lagrangian applies Newton's method directly to an unconstrained minimization problem. The stationary condition of the minimization problem is defined as:

$$\begin{aligned}\nabla_z\mathcal{L}_\rho(z,\lambda,\mu) =& Hx + h + \\ &+ G^T\left[\lambda + \rho(Gx-g)\right] \\ &+ C^T\left[\mu + \rho I_\mu(Cx-c)\right] = 0\end{aligned} \tag{14}$$

And the Gauss-Newton hessian (omit any second-order information of the constraints) of augmented Lagrangian can be constructed as:

$$H_{gn} = H + \rho G^T G + \rho C^T I_\mu C, \tag{15}$$

And the regularized Newton step is calculated as

$$\Delta z = -(H_{gn} + \epsilon I)^{-1}(\nabla_z\mathcal{L}_\rho(z,\lambda,\mu)) \tag{16}$$

As the penalty weight ρ appears in the hessian, it is clear that the hessian becomes increasingly ill-conditioned as the penalty increases, degrading the quality of the Newton step.

III. Primal-Dual Augmented Lagrangian Methods

A. Formulation

The primal-dual augmented Lagrangian method seeks to solve the ill-conditioning issue aroused in the primal method by replacing part of the augmented Lagrangian with extra dual variables $y \in \mathbb{R}^p$ and $w \in \mathbb{R}^q$. Let

$$y = \lambda + \rho(Gz - g) \tag{17}$$
$$w = \mu + \rho I_\mu(Cz - c) \tag{18}$$

The stationary condition (14) thus can be rewritten as

$$\nabla_z\mathcal{L}_\rho(z,\lambda,\mu) = Hz + h + G^T y + C^T w = 0, \tag{19}$$

Rearranging (17) and (18), the optimality conditions for the unconstrained minimization of the augmented Lagrangian can be written as the following system of equations:

$$Hz + h + G^T y + C^T w = 0 \tag{20}$$
$$Gz - g + \frac{1}{\rho}(\lambda - y) = 0 \tag{21}$$
$$I_\mu(Cz - c) + \frac{1}{\rho}(\mu - w) = 0 \tag{22}$$

Note that the rearrangement (21) and (22) is performed strategically so that only the reciprocal of the penalty weight shows up.

The Newton step for the system of equations above can be defined as the following:

$$\begin{bmatrix}\Delta z\\ \Delta y\\ \Delta w\end{bmatrix} = -\begin{bmatrix}H+\epsilon I & G^T & C^T\\ G & -\frac{1}{\rho}I & 0\\ I_\mu C & 0 & -\frac{1}{\rho}I\end{bmatrix}^{-1} r_{gn} \tag{23}$$

where ϵI is a diagonal regularization for the primal variables; r_{gn} is the residual vector defined as the following:

$$r_{gn} = \begin{bmatrix}Hz+h+G^Ty+C^Tw\\ Gz-g+\frac{1}{\rho}(\lambda-y)\\ I_\mu(Cz-c)+\frac{1}{\rho}(\mu-w)\end{bmatrix} \tag{24}$$

Compared to the primal method, this system does not become ill-conditioned as the penalty weight increases, because the weight does not appear in the KKT matrix. Only the reciprocal (i.e. $-\frac{1}{\rho}$) shows up at the diagonal entries associated with the dual variables serving as a dual regularization naturally. By adding ϵI to the cost hessian, the KKT matrix is always invertible. And thus, we do not impose any requirement on the rank of the constraint gradient.

B. Symmetric KKT matrix

Though the KKT matrix of (23) has a significantly better numerical condition, it is not symmetric and thus slow to factorize. To construct a symmetric quasi-definite matrix, the stationary condition (20) is revised.

To facilitate the derivation, a lemma and its proof are presented first.

Lemma 1. *Given a vector $\mu \in \mathbb{R}^q_{\geq 0}$, a selection matrix I_μ that follows the definition of* (9)*,*

$$I_\mu \mu = \mu \tag{25}$$

Proof: Suppose for an index n of the inequality constraint set $\mathbf{C}$, one sufficient condition for $I_\mu(n,n)$ being 1 is $\mu_n > 0$. As $\mu_n \in \mathbb{R}_{\geq 0}$,

$$I_\mu(i,i)\mu_n = \begin{cases}\mu_n & \mu_n > 0\\ 0 & \mu_n = 0\end{cases} \tag{26}$$

And thus, it proves the lemma.

In addition, as the matrix I_μ has either 0 or 1 on the diagonal, it is easy to verify that:

$$I_\mu^2 = I_\mu \tag{27}$$

Following (25) and (27), the multiplication of the selection matrix I_μ and the dual variable w of the inequality constraints can be written as

$$\begin{aligned}I_\mu w &= I_\mu \mu + \rho I_\mu^2(Cz-c)\\ &= \mu + \rho I_\mu(Cx-c)\\ &= w\end{aligned} \tag{28}$$

which is equal to the dual variable itself.

With (28), we are ready to revise the stationary condition of the primal-dual augmented Lagrangian to be the following:

$$Hz + h + G^Ty + (I_\mu C)^T w = 0 \tag{29}$$

And the resulting Newton step becomes the following:

$$\begin{bmatrix}\Delta z\\ \Delta y\\ \Delta w\end{bmatrix} = -\begin{bmatrix}H+\epsilon I & G^T & (I_\mu C)^T\\ G & -\frac{1}{\rho}I & 0\\ I_\mu C & 0 & -\frac{1}{\rho}I\end{bmatrix}^{-1} r_{gn} \tag{30}$$

C. Termination criteria

We define the primal residual r_{primal} and the dual residual r_{dual} as:

$$r_{primal} = \begin{bmatrix}Gz-g\\ max(Cz-c,0)\end{bmatrix} \tag{31}$$

$$r_{dual} = Hz + h + G^T\lambda + C^T\mu \tag{32}$$

The proposed method stops when the norms of the residuals are less than some pre-defined tolerances $\epsilon_{primal} > 0$ and $\epsilon_{dual} > 0$, namely:

$$||r_{primal}||_\infty \leq \epsilon_{primal}, \quad ||r_{dual}||_\infty \leq \epsilon_{dual} \tag{33}$$

And tolerances are set as:

$$\epsilon_{primal} = \epsilon_{primal\_abs} + \epsilon_{rel}||r_{primal}||_\infty \tag{34}$$

$$\epsilon_{dual} = \epsilon_{dual\_abs} + \epsilon_{rel}||r_{dual}||_\infty \tag{35}$$

$$\tag{36}$$

IV. Numerical Experiment

In this section, we will compare our solution with other existing methods discussed above over a trajectory optimization problem without inequality constraints. Specifically, a trajectory optimization problem of a 4D double integrator is benchmarked against OSQP [9] that uses the ADMM, HPIPM [10] that uses the interior-point method, and the proposed method.

A. System dynamics

We consider solving an instant MPC problem of stabilizing a 4D double-integrator. The horizon is then receded by altering the numerical values of the corresponding entries when new information is available. The state $x_t \in \mathbb{R}^8$ is a column vector representing the positions and velocities at time t along four orthogonal axes. The control input $u_t \in \mathbb{R}^4$ represents the acceleration at time t along the four axes. The continuous dynamics is discredited with the second-order Runge-Kutta method; see [11] for details. And the resulting discrete dynamics is defined as:

$$x_{t+1} = Ax_t + Bu_t \tag{37}$$

$$A = \begin{bmatrix} I_{4\times 4} & \Delta t I_{4\times 4} \\ \mathbf{0} & I_{4\times 4} \end{bmatrix}, \; B = \begin{bmatrix} \Delta t^2 I_{4\times 4} \\ \Delta t I_{4\times 4} \end{bmatrix} \tag{38}$$

B. Parameter selection

We use quadratic costs for states and inputs. The state costs $Q = diag([\mathbf{1}_{1\times 4}, \frac{1}{10}\mathbf{1}_{1\times 4}])$, $R = diag([.3, .3, .3, .3])$ are set for all knots. Both the absolute primal tolerance ϵ_{primal_abs} and the absolute dual tolerance ϵ_{dual_abs} are set to be $1e^{-6}$. The relative tolerance ϵ_{rel} is chosen to be $1e^{-12}$. The penalty weight ρ starts from 10 and is enlarged by 2 every iteration. The overall trajectory contains 50 knots in total. With the sampling time Δt chosen to be 0.1s, the total duration is 5s.

C. Results

The absolute solving time of different methods on the problem described above is summarized in table I. And figure 1 shows the normalized solving time with respect to the absolute solving time of OSQP.

TABLE I
SOLVE TIME

	OSQP	PDAL	HPIPM
Time [ms]	0.459	0.242	0.043

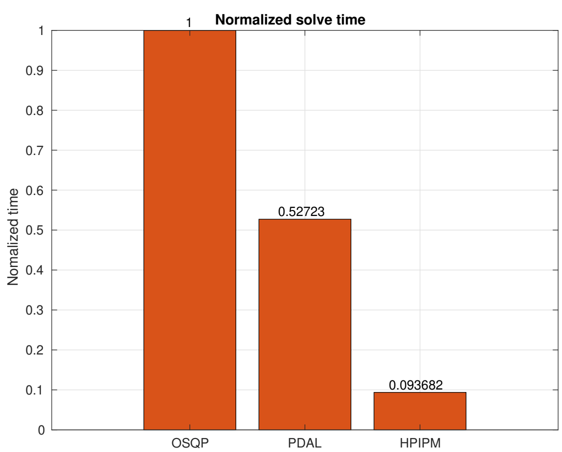


Fig. 1. Normalized solve time

The proposed method outperforms the OSQP by almost a factor of two, but is still significantly slower than HPIPM. We speculate that it is because of the BLASFEO [12], a linear algebra library co-designed with HPIPM and is highly optimized for embedded optimization. We are re-implementing the proposed method based on BLASFEO.

V. CONCLUSIONS

In this paper, we present a novel primal-dual formulation of the augmented Lagrangian which has a significantly better numerical condition than the primal formulation. Without deliberately optimizing the used linear algebra library, the primal-dual augmented Lagrangian has already beaten some state-of-art solvers and demonstrated great potential in the model predictive control. We will continue improving the numerical implementation of the proposed method and hopefully, more promising results will come out in the future.

ACKNOWLEDGMENT

The author (Zhengyu Fu) would like to thank Dr. Zachary Manchester and Dr. Kevin Tracy for their guidance and generous support throughout the project. Thank you to Dr. Dolan and Ms. Burcin for coordinating RISS and making the program possible. This work is supported by the Carnegie Mellon University (CMU) Robotics Institute Summer Scholars (RISS) program and the Robotic Exploration Lab in The Robotics Institute at CMU.

REFERENCES

[1] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2245–2252.

[2] P. Wolfe, "The simplex method for quadratic programming," *Econometrica*, vol. 27, no. 3, pp. 382–398, 1959. [Online]. Available: http://www.jstor.org/stable/1909468

[3] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, 2011.

[5] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[6] E. John and E. A. Yıldırım, "Implementation of warm-start strategies in interior-point methods for linear programming in fixed dimension," *Computational Optimization and Applications*, vol. 41, no. 2, pp. 151–183, Nov 2008. [Online]. Available: https://doi.org/10.1007/s10589-007-9096-y

[7] *Penalty and Augmented Lagrangian Methods*. New York, NY: Springer New York, 2006, pp. 497–528. [Online]. Available: https://doi.org/10.1007/978-0-387-40065-5_17

[8] J.-P. Dussault, "Numerical stability and efficiency of penalty algorithms," *SIAM Journal on Numerical Analysis*, vol. 32, no. 1, pp. 296–317, 1995.

[9] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020. [Online]. Available: https://doi.org/10.1007/s12532-020-00179-2

[10] G. Frison and M. Diehl, "Hpipm: a high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.

[11] J. Butcher, "A history of runge-kutta methods," *Applied Numerical Mathematics*, vol. 20, no. 3, pp. 247–260, 1996. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0168927495001085

[12] G. Frison, D. Kouzoupis, T. Sartor, A. Zanelli, and M. Diehl, "Blasfeo: Basic linear algebra subroutines for embedded optimization," *ACM Transactions on Mathematical Software (TOMS)*, vol. 44, no. 4, pp. 1–30, 2018.

[13] P. E. Gill and E. Wong, "Sequential quadratic programming methods," in *Mixed Integer Nonlinear Programming*, J. Lee and S. Leyffer, Eds. New York, NY: Springer New York, 2012, pp. 147–224.

RiSS | ROBOTICS INSTITUTE SUMMER SCHOLARS

riss.ri.cmu.edu

Carnegie Mellon University
The Robotics Institute

Carnegie Mellon University
School of Computer Science