ROBOT LEARNING FOR LIFELONG ADAPTATION
IN OPEN-WORLD ENVIRONMENTS

by

Sriram Siva

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Computer Science).

Golden, Colorado

Date ________________________

Signed: ________________________

Sriram Siva

Signed: ________________________

Dr. Hao Zhang
Thesis Advisor

Golden, Colorado

Date ________________________

Signed: ________________________

Dr. Iris Bahar
Department Head and Professor
Department of Computer Science

ABSTRACT

The use of robotic systems has become increasingly vital in modern society, owing to their superior precision, reliability, effectiveness, and flexibility in performing complex and demanding tasks, such as search and rescue operations, autonomous driving, planetary and subterranean exploration, human-robot collaboration, and homeland defense. With the integration of artificial intelligence and machine learning, the potential applications for robotic systems are expanding even further, enabling robots to perform tasks beyond their pre-programmed capabilities. As a result, robotic systems are now ubiquitous in our daily lives, providing greater convenience and ease of use, and offering new possibilities for innovation and growth.

Robot learning is critical for enabling intelligent robotic systems to operate successfully. By incrementally learning from their experiences or interactions with their environment, robots can improve their understanding of their surroundings and capabilities. However, in open-world environments, operations are conducted in highly unstructured, dynamic, and uncertain surroundings, posing significant challenges for robot learning. Accordingly, there is a need for robot learning that enables lifelong adaptation allowing robots to adjust their capabilities to continue to perform operations successfully and efficiently in open-world environments.

This dissertation focuses on addressing the challenges of robot lifelong operations in open-world environments by developing robot learning methods that enable adaptation on the fly. The research introduces regularized optimization-based and deep-learning methods that first make the robot robust and later enable it to adapt to long-term autonomy challenges with multiple sensor modalities. In addition, the dissertation proposes robot learning methods to adapt not only to changes in its surroundings but also within itself, such as changes in its capabilities. The proposed approaches have been successfully integrated into real-world robotic systems and demonstrate efficient solutions to real-world autonomy challenges through demonstrations in real-world operations.

TABLE OF CONTENTS

ABSTRACT iii

LIST OF FIGURES xi

LIST OF TABLES xvii

ACKNOWLEDGMENTS xviii

DEDICATION xix

CHAPTER 1 INTRODUCTION 1

1.1 Background 2

1.1.1 Open-World Environments 3

1.1.2 Robot Learning 4

1.1.3 Robot Lifelong Adaptation 5

1.2 Objective 6

1.3 Main Contributions 6

1.3.1 Robot Learning for Lifelong Adaptation to External Environments 7

1.3.2 Learning Capability Awareness for Lifelong Adaptation in open-world Environments 7

1.4 Structure of the Dissertation 8

CHAPTER 2 OMNIDIRECTIONAL MULTISENSORY PERCEPTION FUSION FOR LONG-TERM PLACE RECOGNITION 9

2.1 Abstract 9

2.2 Introduction 10

2.3 Related Work 12

2.3.1 Scene Matching for Place Recognition 12

2.3.2 Representation in Long-Term Place Recognition 12

2.4 The FOMP Approach 13

2.4.1 Problem Formulation 14

2.4.2 Learning Discriminative Views 15

2.4.3 Learning Discriminative Modalities 17

2.4.4 Omnidirectional Multisensory Place Recognition 18

2.4.5 Optimization Algorithm 18

2.5 Omnidirectional Multisensory Dataset 21

2.6 Experiments 22

2.6.1 Experimental Setup 22

2.6.2 Results at Different Times of a Day 24

2.6.3 Results across Different Seasons 25

2.6.4 Results on Multi-directional Place Recognition 27

2.7 Summary 28

CHAPTER 3 ROBOT PERCEPTUAL ADAPTATION TO ENVIRONMENT CHANGES FOR LONG-TERM HUMAN TEAMMATE FOLLOWING 29

3.1 Abstract 29

3.2 Introduction 30

3.3 Related Work 34

3.3.1 Long-Term Autonomy 35

3.3.2 Robot Adaptation 36

3.3.3 Human Following 36

3.4 The ROPA Approach 37

3.4.1 Multimodal Sensor Fusion 38

3.4.2 Perception Calibration 39

3.4.3 Multisensory Robot Perceptual Adaptation 41

3.4.4 Optimization Algorithm 41

3.5 Dataset and Implementation 46

3.5.1 The PEAC Dataset 46

3.5.2 Implementation 48

3.6 Experiments 49

3.6.1 Scenario I (Mine-Entering-Exiting) 49

3.6.2 Scenario II (Indoor-Outdoor-Traveling) 51

3.6.3 Scenario III (All-Day-Following) 52

3.7 Discussion 54

3.7.1 High-Speed Processing 54

3.7.2 Feature Modality Analysis 55

3.7.3 Hyperparameter Selection 55

3.8 Summary 56

CHAPTER 4 ROBOT ADAPTATION TO UNSTRUCTURED TERRAINS BY JOINT REPRESENTATION AND APPRENTICESHIP LEARNING 58

4.1 Abstract 58

4.2 Introduction 59

4.3 Related Work 61

4.3.1 Terrain Classification and Characterization 61

4.3.2 Robot Adaptation 62

4.4 Approach 64

4.4.1 Representation Learning for Terrain Classification 64

4.4.2 Formulation of Joint Representation and Apprenticeship Learning for Unstructured Terrain Adaptation 65

4.5 Optimization Algorithm 68

4.6 Experiments 71

4.6.1 Experiment Setup and Dataset 71

4.6.2 Implementation 73

4.6.3 Result on Familiar Terrains 74

4.6.4 Result on Unfamiliar Unstructured Terrains 77

4.6.5 Discussion 78

4.7 Summary 79

CHAPTER 5 NAUTS: NEGOTIATION FOR ADAPTATION TO UNSTRUCTURED TERRAIN SURFACES 80

5.1 Abstract 80

5.2 Introduction 81

5.3 Related Work 83

5.4 Approach 86

5.4.1 Learning Policy Prediction Models 86

5.4.2 Robot Negotiation for Terrain Adaptation 87

5.4.3 Optimization Algorithm 89

5.5 Proof of Convergence for the Optimization Algorithm 91

5.6 Experiments 93

5.6.1 Experimental Setup 93

5.6.2 Navigating over Dynamic Uncertain Grass Terrain 96

5.6.3 Navigating on Unseen Unstructured Forest Terrain 97

5.7 Summary 99

CHAPTER 6 CAPABILITY-AWARE PERCEPTUAL ADAPTATION FOR ROBUST GROUND NAVIGATION IN UNSTRUCTURED OFF-ROAD ENVIRONMENTS 100

6.1 Introduction 100

6.2 Related Work 104

6.2.1 Robot Terrain Navigation 104

6.2.2 Robot Perceptual Adaptation 105

6.3 Approach 106

6.3.1 Learning Capability Awareness in Perception 107

6.3.2 Learning Robot Dynamics Model 110

6.3.3 Robot Executing Perceptual Adaptation during Navigation 112

6.4 Optimization Algorithm 113

6.5 Experiments 116

6.5.1 Experimental Setups 117

6.5.2 Navigating with Obscured Sensors 118

6.5.3 Navigating with Failed Sensors 121

6.6 Conclusion 122

CHAPTER 7 ENHANCING CONSISTENT GROUND MANEUVERABILITY BY ROBOT ADAPTATION TO COMPLEX OFF ROAD TERRAINS 123

7.1 Abstract 123

7.2 Introduction 124

7.3 Related Work 126

7.4 Approach 127

7.4.1 Problem Formulation for Terrain-Aware Navigation 127

7.4.2 Consistent Navigational Behavior Generation 130

7.4.3 Optimization Algorithm 132

7.5 Proof of Convergence for the Proposed Optimization Algorithm 133

7.6 Experiments 137

7.6.1 Navigating over Individual Types of Unstructured Terrain 138

7.6.2 Navigating over Complex Off-road Unstructured Terrains 139

7.6.3 Discussion 141

7.7 Implementation of the Algorithm and Robot Software System 143

7.7.1 Algorithm Implementation and Training/Execution Procedures 143

7.7.2 Robot Software System Implementation 145

7.8 Assumptions and Limitations 146

7.9 Conclusion 148

CHAPTER 8 SELF-REFLECTIVE TERRAIN-AWARE ROBOT ADAPTATION FOR CONSISTENT OFF-ROAD GROUND NAVIGATION 149

8.1 Introduction 149

8.2 Related Work 153

8.2.1 Control Methods for Ground Navigation 153

8.2.2 Learning-Based Methods for Navigation 153

8.2.3 Self-Reflective Robot Adaptation 155

8.3 Approach 155

8.3.1 Terrain-Aware Ground Navigation 157

8.3.2 Self-Reflection for Consistent Navigational Behavior Generation 160

8.4 Optimization Algorithm 163

8.5 Experiments 170

8.5.1 Experimental Setups 170

8.5.2 Robot Navigation over Individual Unstructured Off-road Terrains 172

8.5.3 Robot Adaptation to Terrain Transitions 174

8.5.4 Robot Self-Reflective Adaptation over Unstructured Terrains 176

8.5.5 Discussion 181

8.6 Conclusion 182

CHAPTER 9 CONCLUSION 184

9.1 Future Directions 188

9.1.1 Semantic Aware Active Perception for Enhanced Terrain Awareness 188

9.1.2 Proactive High-Speed Autonomy 188

9.1.3 Heterogenous Multi-Robot Collaborative Autonomy 189

REFERENCES 190

APPENDIX COPYRIGHT PERMISSION 217

A.1 IEEE, RSS, CoRL and Sage Reuse Policies 217

A.2 Permission from Co-Authors 218

LIST OF FIGURES

Figure 1.1 Motivating scenarios of robot lifelong adaptation in unstructured environments. When operating in these environments, mobile robots will encounter a complex combination of discrete or continuous and gradual or sudden changes in the environment and the robot itself. In addition, these environments have varying characteristics which cannot be modeled into the robot before deployment. Autonomous robots must, therefore, successfully adapt their perception, behaviors, and decision-making capabilities to achieve robust autonomy in such environments. 2

Figure 1.2 Overview of the research proposed in this dissertation. 6

Figure 2.1 Illustration of the weight matrix $\mathbf{W}$. 15

Figure 2.2 Illustration of the proposed cone-structured sparsity-inducing norm for learning discriminative views. Figure 2.2(a) illustrates the 3D structure of our proposed cone-structured sparsity-inducing norm and Figure 2.2(b) illustrates the corresponding unwrapped structure of the same sparsity-inducing norm. . 17

Figure 2.3 Results over Route-A across different times of a day (i.e., morning versus evening). 23

Figure 2.4 Results over Route-B across different times of a day (i.e., morning versus evening). 23

Figure 2.5 Results over the Route-B dataset across different seasons (i.e., Summer versus Fall). 25

Figure 2.6 Experimental results on view and modality importance using the Route-A dataset across various times of a day. 26

Figure 2.7 Results over the Route-B dataset across different times of a day (i.e., morning versus evening). 26

Figure 2.8 Experimental results on view and modality importance using the Route-B dataset across different seasons. 27

Figure 2.9 Results of bidirectional place recognition when a car drives through the same place from different directions in different seasons. In the legend of precision-recall curves, "*For*" indicates the FOMP approach is trained using forward-direction data only, and "*Back*" indicates the approach is trained using backward-direction data only. Testing is performed using data collected from both directions. 27

Figure 3.1 Motivating examples of robot perceptual adaptation in human teammate following applications. When a robot follows a human during a long-term operation (e.g., search and rescue), the robot requires the capability of perceptual adaptation to adapt to fast changes (e.g., when moving from a dark tunnel to a bright open area shown in the top row) and long-term changes (e.g., different times of the day shown in the bottom row) and in order to avoid perception failures. The problem of robot adaptation to fast and dramatic environment changes has not yet been well addressed. For example, given a stream of color and depth data as the input, existing methods of human detection cannot adaptively choose sensing modalities and often lose the person when the robot travels from a dark mine to a bright open area with fast and dramatic lighting changes. 30

Figure 3.2 Illustration of the scenarios in which the PEAC dataset was collected for the task of human teammate following. 45

Figure 3.3 Qualitative and quantitative results on Scenario I (Mine-Entering-Exiting). The top row depicts an example of the qualitative results from the robot's viewpoint when it follows a human subject to navigate into a dark mine drift from a bright open area. The bottom row illustrates the importance of color-depth sensor modalities learned and adapted by ROPA. 46

Figure 3.4 Qualitative and quantitative results on Scenario II (Indoor-Outdoor-Traveling). The top row illustrates an example of the recognition results from the robot's viewpoint when it follows the human to navigate from the inside to the outside of a building. The bottom row shows the importance of color-depth sensor modalities learned and adapted by our ROPA algorithm. 50

Figure 3.5 Qualitative and quantitative results on Scenario III (All-Day-Following). The top row illustrates an example of the qualitative results from the robot's viewpoint when it follows a subject to navigate in a campus environment from noon till late evening with dramatic long-term lighting changes. The bottom row shows the importance of color-depth sensor modalities learned and adapted by ROPA. 52

Figure 3.6 Experimental results on feature modality analysis during the training phase. . . 54

Figure 3.7 Experimental results on hyperparameter analysis during the training phase. . . 55

Figure 4.1 Motivating scenarios of robot adaptation to unstructured terrains. When mobile robots operates in a field environment during disaster response, the robots need the essential capability of navigating over a variety of terrains, such as grass, rock, snow, pebble, and mud, which often cannot be accurately structured before onsite robot deployment. In this paper, we propose the TRAL method to integrate representation learning and apprenticeship learning under a unified mathematical framework and estimate terrain feature importance to enable robust robot adaptation to unstructured terrains. 59

Figure 4.2 Illustration of the proposed TRAL approach for joint representation and apprenticeship learning under the unified regularized optimization framework, with the regularization terms $\|\mathbf{V}\|_R$ and $\|\mathbf{V}\mathbf{U}\|_A$ to identify the discriminative feature modalities. 66

Figure 4.3 Instances of the five terrain types used in the experiments. The color images are observed by the camera installed in front of the mobile robot when navigating over these unstructured terrains. 73

Figure 4.4 Quantitative results of RMSE on ground velocity obtained by TRAL in the scenarios of navigation over *familiar* unstructured terrains. Comparison with previous apprenticeship learning methods and our baseline methods is also presented. The results of each method over each type of terrains are presented as a boxplot that depicts the distribution of the results from all testing instances based on a five number summary (i.e., minimum, first quartile, median, third quartile, and maximum). 75

Figure 4.5 Quantitative results of RMSE on ground velocity obtained by TRAL in the scenarios of navigation over *unfamiliar* unstructured terrains. Comparison with previous apprenticeship learning methods and our baseline methods is also presented. 77

Figure 4.6 Normalized importance of terrain feature modalities with respect to terrain classification (Figure 4.6(a)) and behavior generation (Figure 4.6(b)). 79

Figure 5.1 Robots operating in dynamic, unstructured environments often generate sub-optimal behaviors leading to inefficient robot traversal or even navigation failure. For example, in tall grass terrain, robots consider the terrain as an obstacle. Terrain negotiation allows robots to explore different navigation policies to determine the optimal combination for successful and efficient navigation in unknown terrains. In this example, the robot initially treats tall grass as an obstacle but simultaneously explores a max speed policy. The robot then quickly learns that the max speed policy improves efficiency by traversing across tall grass, learning to give more importance to max speed over obstacle avoidance. 81

Figure 5.2 Overview of our proposed NAUTS approach for robot negotiation to adapt over unstructured terrains. Illustrated is the learning performed by our approach during the training phase. The module in the yellow box illustrates robot negotiation during the execution stage. 85

Figure 5.3 A shallow GP is designed to implement our prediction model $f_{\mathbf{w}^i}$. 86

Figure 5.4 The tall grass terrain used in our experiments and the qualitative results over this terrain. 94

Figure 5.5 The forest terrain used in our experiments and the qualitative results over this terrain. 97

Figure 6.1 Motivational images to develop capability awareness in robot navigation. As seen from the figure, robots navigating in unstructured environments often encounter perceptual challenges like sensor obscurations or failures. Such challenges may cause errors in robot localization and inaccuracy or even failed navigation. Therefore, the capability of ground robots to be capability-aware and adapt to these perceptual challenges is essential to the successful completion of the navigation task. 101

Figure 6.2 Overview of our proposed approach. From the left, the first module of the figure illustrates the generative network used to represent the generator model, the second module illustrates the discriminative network used to represent the discriminator network and the third module shows the shallow network used to represent the dynamics model of the robot. 107

Figure 6.3 Unstructured terrains in the wild used for our experimentation. 116

Figure 6.4 The three different scenarios of sensor obscurations used in our experiments. . 116

Figure 7.1 Off-road environments such as forests are unstructured and exhibit a variety of characteristics, including changing terrain types and slopes. When ground robots are deployed in these environments, their actual behaviors often do not match the expected behaviors, e.g., due to wheel slip. The inconsistency often causes slower traversal time or errors in robot state estimation. Therefore, the capability of consistent behavior generation is essential for maneuverability while ground robots navigate over unstructured off-road terrains. 124

Figure 7.2 Overview of our proposed approach. 129

Figure 7.3 Individual types of unstructured terrain used in the experiments. 138

Figure 7.4 Complex unstructured off-road terrains used in the experiments. 140

Figure 7.5 Hyperparameter Analysis. 141

Figure 7.6 Seq. length. 142

Figure 7.7 Convergence. 142

Figure 7.8 Relative importance of the feature modalities estimated by our approach to enable consistent off-road ground navigation. 143

Figure 7.9 Overview of the ROS-based software architecture implemented on our physical robots for performing consistent behavior generation. The yellow box shows our approach and the remaining modules that work with the proposed approach in our software architecture are shown by gray boxes. 145

Figure 7.10 Multimodal features used to characterized terrain during ground navigation. . . 146

Figure 8.1 A motivating scenario of self-reflective terrain-aware robot adaptation for generating consistent robot navigational behaviors. When robots operate in unstructured off-road environments, the actual behaviors often do not match their expected behaviors, due to changes in the characteristics of terrains and the robots themselves. Therefore, the capability of robot adaptation for consistent behavior generation is essential for maneuverability over unstructured off-road terrains. 150

Figure 8.2 Overview of our proposed method for self-reflective terrain-aware robot adaptation to enable ground robots to adapt their navigational behaviors to unstructured off-road terrains, and perform self-reflection to generate consistent navigational behaviors in the unified constrained regularized optimization framework. 156

Figure 8.3 Multi-modal features extracted from the robot's onboard camera and LiDAR to characterize unstructured off-road terrains in our experiments. 170

Figure 8.4 Individual types of terrains used in the experiments. From left to right: concrete, grass, large-rock, gravel, and snow. 172

Figure 8.5 The off-road ground navigation circuit consists of different types of terrains (such as grass, gravel, large-rock, mud, and sand) as well as diverse terrain transitions. The size of the circuit is 10 x 21 meters with terrain slopes varying between $0 - 30°$. 173

Figure 8.6 Various terrain tracks used to evaluate the self-reflection capability of the robot. From left to right: mixed terrain I (MT-I), mixed terrain II (MT-II), forest, reduced-traction snow-grass (RT-Snow-Grass), reduced-traction mud-snow (RT-Snow-Mud), reduced-traction hill (RT-Hill), reduced-traction forest (RT-Snow-Forest), reduced-traction payload snow-grass (RT-P-Snow-Grass) and reduced-traction payload hill terrain (RT-P-Hill). 176

Figure 8.7 Analysis of our approach based on its various parameters. 180

Figure 9.1 Research accomplishments on robot learning for lifelong adaptation in open-world environments. 184

Figure A.1 Copyright permission from Dr. Hao Zhang. 218

Figure A.2 Copyright permission from Dr. Maggie Wigness. 219

Figure A.3 Copyright permission from Dr. John Rogers. 219

Figure A.4 Copyright permission from Long Quang. 220

LIST OF TABLES

Table 3.1 Comparison of average accuracy over all robot operation scenarios in the PEAC dataset . . . 50

Table 4.1 Mean and variance of RMSE and ER on ground velocity obtained by TRAL and previous/baseline methods over familiar unstructured terrains. . . . 76

Table 4.2 Mean and variance of RMSE and ER on ground velocity obtained by TRAL and previous/baseline methods over unfamiliar unstructured terrains. . . . 78

Table 5.1 Quantitative results for scenarios when the robot traverses over dynamic, uncertain grass terrain. . . . 96

Table 5.2 Quantitative results for scenarios when the robot traverses over unseen dynamic, unstructured off-road forest terrain. . . . 98

Table 6.1 Quantitative results for scenarios when the robot encounters sensor obscurations. Successful runs (with no navigation failures) are used to calculate the metrics of traversal time and adaptation time. . . . 118

Table 6.2 Quantitative results for scenarios when the robot encounters sensory failures. Successful runs (with no navigation failures) are used to calculate the metrics of traversal time and adaptation time. . . . 120

Table 7.1 Quantitative results based on ten runs for scenarios when the robot traverses over *individual types of unstructured terrain*. Successful runs (with no failures) are used to calculate the metrics of traversal time, inconsistency and jerkiness. . . . 139

Table 7.2 Quantitative results for scenarios when the robot traverses over *complex unstructured off-road terrain* shown in . . . 140

Table 8.1 Definition of the key variables used in our approach. . . . 156

Table 8.2 Quantitative results for scenarios when the robot traverses over the individual types of unstructured terrain shown in . . . 172

Table 8.3 Quantitative results based on ten runs for scenarios when the robot navigates over the various terrain transitions shown in . . . 174

Table 8.4 Quantitative results based on ten runs for the terrain scenarios shown in . . . 178

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my parents. Their constant love and support have been an endless source of inspiration for me. I dedicate this work to them.

Next, I want to take a moment to express my heartfelt gratitude to my advisor, Hao Zhang, for his exceptional guidance and mentorship. His expertise and passion for research have instilled in me a deep curiosity for scientific inquiry and the value of patience. Not only is he a respected researcher in the robotics community, but also a kind and compassionate person who sets a great example for me as I embark on my own career.

I would like to thank my committee members, Dr. Andrew Petruska, Dr. Tom Williams, and Dr. Hua Wang, whose support and collaboration have been invaluable to the success of my research. I am equally thankful to my colleagues in the Human-Centered Robotics Lab, whose camaraderie and assistance have created a productive and enjoyable research environment.

In addition, I would like to express my appreciation to my collaborators at US DEVCOM Army Research Lab, Dr. Maggie Wigness, Dr. John Rogers, Long Quang, and Eric Spearo, for their valuable advice, and technical support that helped me immensely to advance my research.

I would also like to acknowledge and thank my undergraduate teachers, Dr. Harish, Dr. Balakrishnan Shankar, Dr. Anand Ramachandran, and Madhu Meher, for teaching me the fundamentals of science and engineering, which laid the foundation for my doctoral research.

Finally, I am grateful to my fiance, Mekha Menon, for being a pillar of strength and supporting me through difficult times.

Dedicated to my parents Mrs. Sasikala Shanmugam & Mr. Siva Shanmugam.

CHAPTER 1

INTRODUCTION

This dissertation focuses on developing robot learning methods that enable robots to adapt to challenges posed by open-world environments. The motivation is to create highly intelligent robots that can successfully complete field missions in complex and congested environments by continuously improving and adapting their perception, localization, mapping, and navigation abilities with minimal human intervention.

Mobile robots have the unparalleled potential to revolutionize a variety of applications, including disaster response [1], subterranean and planetary exploration [2], homeland defense [3], autonomous driving [4], and more. In order to be successful in these applications, robots must be able to sense and operate with limited previous knowledge of the environment and with minimal human interventions or external supporting infrastructure. In addition, operation in several of these environments entails limited bandwidth, often making teleoperation impossible and fully autonomous operations necessary. Essentially, robots operations should not be limited by the environment. Therefore, mobile robots must possess a high level of perception and reasoning capabilities, as well as the ability to make safe decisions while navigating in open-world environments, in order to promise effective and efficient mission outcomes.

However, autonomous operations over an extended period of time are extremely challenging for mobile robots, as they must adapt to unpredictable changes in lighting, weather, vegetation, and other environmental conditions. As can be seen from Figure 1.1, this challenge is further compounded in open-world environments, where robot perception, navigation, decision-making, localization, and mapping are further complicated by the dynamic and uncertain nature of the surroundings which cannot be modeled in advance. Additionally, these robots must not only adapt to external changes but also be resilient to internal changes, such as wear and tear, sensor failure, or changes in robot payload.

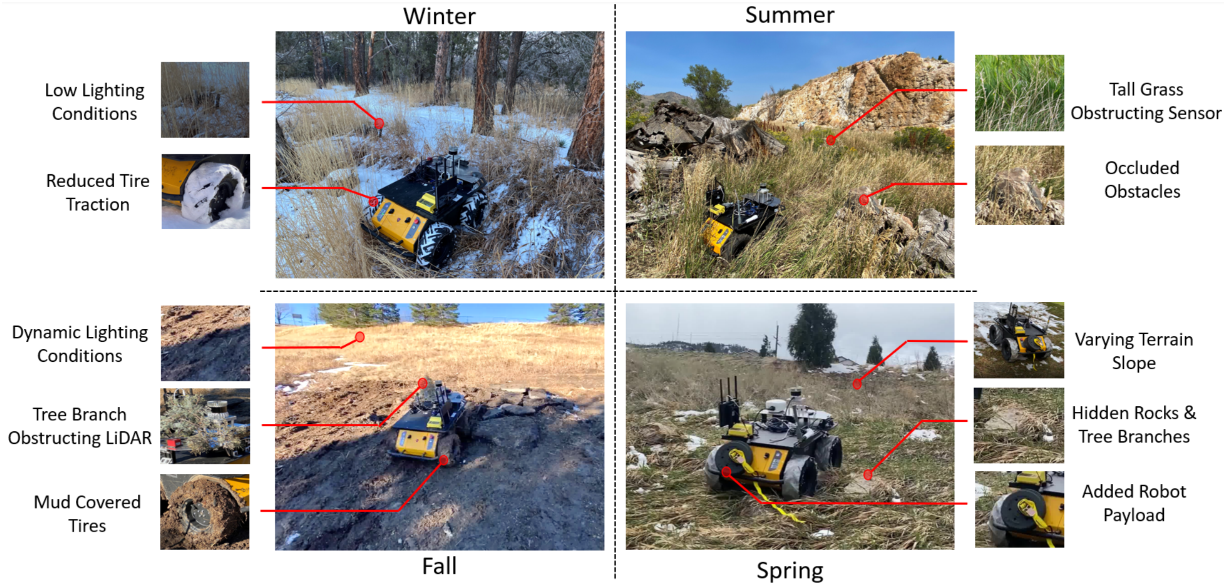


Figure 1.1 Motivating scenarios of robot lifelong adaptation in unstructured environments. When operating in these environments, mobile robots will encounter a complex combination of discrete or continuous and gradual or sudden changes in the environment and the robot itself. In addition, these environments have varying characteristics which cannot be modeled into the robot before deployment. Autonomous robots must, therefore, successfully adapt their perception, behaviors, and decision-making capabilities to achieve robust autonomy in such environments.

In this dissertation, I make a significant contribution to the field of computer science by proposing novel methods for robot learning based on regularized optimization and deep learning. These methods enable robots to learn and adapt in open-world environments, leading to successful and efficient lifelong robot operations. Specifically, my proposed methods address two critical challenges of robot lifelong adaptation: i) robot learning to adapt to external surroundings and ii) learning self-adaptation through capability awareness. By developing and advancing the tools of regularization techniques and deep learning, my proposed methods provide real-time adaptable and scalable solutions for these challenges.

1.1 Background

The background for this dissertation is presented in this section, including *open-world environments*, *robot learning*, and *robot lifelong adaptation*.

1.1.1 Open-World Environments

An open-world environment in robotics refers to real-world physical spaces that are unpredictable and unstructured, where there is no known set of rules [5, 6]. A robot in such an environment must navigate and interact in a flexible and adaptable way, perceiving and understanding the environment, making decisions based on its objective, and adjusting its behaviors accordingly. The robot must also handle unexpected situations, obstacles, and changes in the environment. In essence, the world the robot operates in is *open* because it contains objects that are not described in its internal representation [7].

Robot operation in open-world environments presents three main challenges: long-term variability, unstructured surroundings, and changes within the robot. Long-term variability refers to changes in lighting conditions, weather, vegetation, and other visual appearances that occur over weeks, months, or even years. For example, autonomous driving may encounter significant changes in weather, illumination, and vegetation conditions during different seasons. The challenges associated with unstructured surroundings include the uncertain and complex nature of the environment, where mobile robots encounter a combination of gradual or sudden changes in both the environment and the robot itself. These environments have varying characteristics that cannot be modeled before deployment. Changes within the robot arise from its extended operation in open-world environments, where it experiences wear and tear, varying payload, and perceptual changes like sensor failure or obscuration. These challenges often occur in multiple combinations and adversely affect robot operations. In robot perception, for example, these challenges may lead to uncertainty in the robot's understanding of the environment.

In summary, open-world environments present a unique set of challenges for robotics, requiring robots to be able to intelligently perceive, navigate, interact with objects, respond to unexpected events, and also operate robustly in the face of unexpected events, such as sensor failures or changes in the environment in a way that is safe and effective.

1.1.2 Robot Learning

Robot learning is the process of enabling robots to acquire new knowledge, skills, and abilities through experience and interaction with the environment, without being explicitly programmed to do so [8]. It is an interdisciplinary field at the intersection of robotics, artificial intelligence, machine learning, control theory, and cognitive science. Traditionally, robots have been limited to performing specific tasks in controlled environments, such as factory assembly lines. However, this limits the use case of robots both in terms of tasks and environments. Robot learning enables robots to overcome these limitations by learning from their experiences and adapting to new situations. This requires developing algorithms and techniques for perceiving the environment, reasoning about actions, and learning from feedback.

Over years, several different approaches have been proposed under robot learning and all of these approaches generally fall under the machine learning algorithms of supervised robot learning, unsupervised robot learning, and reinforcement learning. In supervised robot learning, a robot is trained on a set of labeled examples to learn how to perform a specific task [9, 10]. The robot learns a mapping from input data to output data to make predictions using neural networks, tools from statistics, or other methods. In unsupervised robot learning, the robot is not provided any labeled examples, rather it learns by discovering patterns and structures in the input data [11, 12]. This approach of robot learning is useful when there exists limited knowledge of the environment in which the robot is operating. Finally, reinforcement learning involves learning through trial and error and feedback in the form of reward or penalties [13, 14]. Specifically, the robot learns to maximize its rewards over time. These approaches have been used alone or in combination to enable robots to learn and generalize to various environments.

However, despite the significant advancements that have been made in robot learning, there are many open challenges that are yet to be addressed. These challenges include developing algorithms that can learn from sparse and noisy data, dealing with uncertainty and ambiguity in open-world environments, and generalizing to unseen and dynamic environments while ensuring the safety and reliability of the robot.

1.1.3 Robot Lifelong Adaptation

While learning in robots involves the process of developing their decision-making models by experience over time, adaptation in robots is their ability to adjust their decision-making abilities on the fly to deal more successfully with a new situation [15]. Adapting to change and novelty is crucial for robots to achieve lifelong operations. Without adaptation, robots will inevitably fail in completing missions and could harm themselves. Lifelong adaptation would enable robots to operate autonomously in open-world environments by continuously changing their perception and behavior models in response to changing conditions.

Early research on robot adaptation focused on high-level behavior-based methods. For instance, [16, 17] used case-based reasoning methods for robot behavioral adaptation in evolving environments. Other methods included online system parametric identification [18], dynamics programming for adaptation [19, 20], and the selection of pre-determined solutions from a library of well-performing ones [21] for controlling complex robots. However, these early methods require significant domain expertise, and most of them rely on the internal states of the robot, failing to learn from high-dimensional sensory data to better understand the environment.

Recently, reinforcement learning has been used successfully in robot behavior adaptation tasks by maximizing rewards during task execution in simulated environments [22–25]. However, this trial-and-error approach is not feasible in real-world scenarios, and the sim-to-real challenge remains a significant bottleneck to implementing simulation-learned models in real-world environments. Other approaches to robot learning for adaptation include transfer learning, online learning, and active learning. Transfer learning involves training the robot on one task and transferring the knowledge to a new, related task [26, 27]. Online learning allows the robot to learn a model of the task in real-time through interaction with the environment [28, 29]. However, online learning can be slow, and catastrophic forgetting can occur [30]. Active learning involves the robot actively selecting the most informative data to learn from [31, 32]. Despite the advances, the challenge of adapting robot perception and behavior models on the fly with changes in both environment and changes within the robot has not been well addressed.

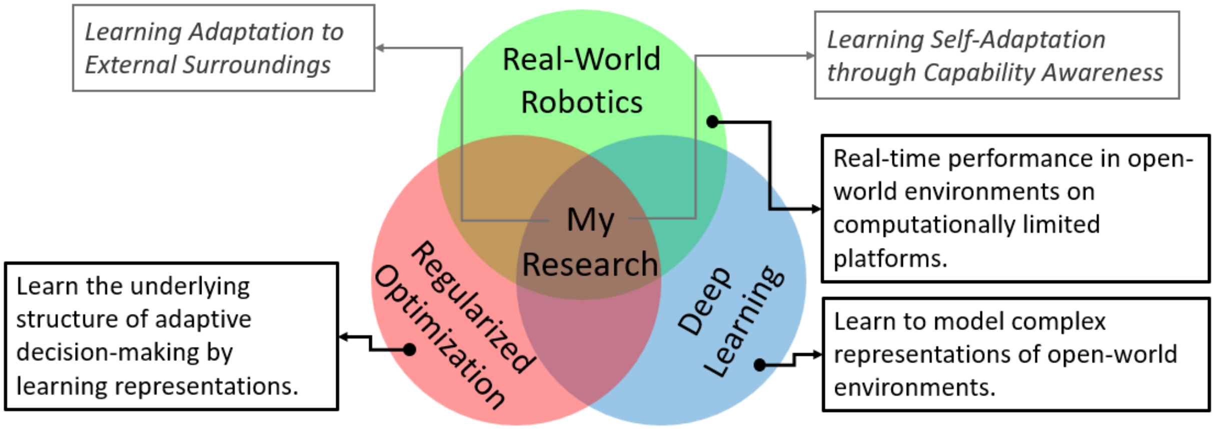

Learning Adaptation to External Surroundings
Learning Self-Adaptation through Capability Awareness
Real-World Robotics
My Research
Regularized Optimization
Deep Learning
Real-time performance in open-world environments on computationally limited platforms.
Learn the underlying structure of adaptive decision-making by learning representations.
Learn to model complex representations of open-world environments.

Figure 1.2 Overview of the research proposed in this dissertation.

1.2 Objective

The objective of this Ph.D. dissertation is to enhance the lifelong adaptation capabilities of robots in order to autonomously and efficiently tackle the challenges of lifelong operation in open-world environments. This dissertation aims to provide practical and effective machine learning-based methods for perception and behavior generation. The research conducted in this dissertation is at the intersection of real-world robotics, regularized optimization, and deep learning, as illustrated in Figure 1.2. Furthermore, The dissertation addresses the challenges of deploying robots in open-world environments by developing robot learning methods that can adapt to changes in both the external surroundings and the robot itself.

1.3 Main Contributions

The main contributions of this dissertation can be divided into two groups. First, I develop methods that enhance lifelong operations in robots via learning adaptation to external environmental changes. Second, one step ahead, I develop learning methods for lifelong operations in robots by learning to adapt not only to environmental changes but also to changes within the robot itself by developing capability awareness. I summarize the main novelties of the proposed robot learning methods for lifelong adaptation in open-world environments as follows:

1.3.1 Robot Learning for Lifelong Adaptation to External Environments

- In Chapter 2, I introduce the *Fusion of Multisensory Perception* (FOMP) approach, which is a novel method for identifying and integrating multimodal data from heterogeneous sensors in different views [9]. The purpose of this approach is to solve the problem of place recognition in long-term autonomy, specifically from omnidirectional multisensory observations.

- In Chapter 3, I introduce a novel, human inspired approach called *RObot Perceptual Adaptation* (ROPA) that enables robots to calibrate their multisensory perception, enabling them to adapt to short- and long-term changes in their environment [10].

- In Chapter 4, I introduce the *Terrain Representation and Apprenticeship Learning* (TRAL) approach, which tackles the problem of robot adaptation to unstructured terrains by simultaneously learning terrain representations and learning from demonstrations within a unified regularized optimization framework [6].

- In Chapter 5, I introduce the novel *Negotiation for Adaptation over Unstructured Terrain Surfaces* (NAUTS) approach for robot adaptation by negotiation for navigating in unstructured terrains. The NAUTS approach allows ground robots to adapt their navigation policies using a negotiation process, enabling them to effectively traverse challenging environments [33].

1.3.2 Learning Capability Awareness for Lifelong Adaptation in open-world Environments

- In Chapter 6, I introduce a novel human-inspired approach to learning capability-aware perceptual adaptation, which allows the robot to adjust to perceptual challenges of sensor failure, and obscuration and successfully navigate in unstructured off-road environments through the learning of a modality invariant trajectory representation in a self-supervised fashion.

- In Chapter 7, I introduce a novel approach for generating consistent behavior in ground robots, enabling their actual behavior to more accurately match expected behaviors while adapting to a variety of unstructured off-road terrain. The approach is based on regularized optimization techniques and presents a significant advancement in the field of robotics, as it addresses a critical need for robots to exhibit consistent behavior while adapting to the various challenges posed by unstructured terrains [34].

- In Chapter 8, a novel approach for generating consistent controls to navigate over unstructured off-road terrains while also being aware of the robots' own navigational capabilities including varying payload, traction, etc.

1.4 Structure of the Dissertation

This dissertation is structured as follows. Chapter 1 introduces this dissertation proposal, including a brief explanation of the background, the objective of this dissertation, and the main contributions. Chapter 2 introduces a representation learning approach that allows robots to perform robust long-term place recognition via enhancing loop-closure detection and achieve lifelong operations. Chapter 3 presents the Robot Perceptual Adaptation (ROPA) method. Chapter 4 and 5 describe two different approaches for achieving all-terrain traversal capability via behavior adaptation. Chapter 6 introduces an adversarial learning method that develops capability awareness in perception. Chapter 7 and 8 introduce methods that enhance terrain navigation via developing capability awareness. Finally, Chapter 9 concludes this dissertation. The chapters in this dissertation are modified from accepted or submitted papers and are noted with the conference, journal, and manuscript author. The dissertation author is the first author of all included chapters.

CHAPTER 2

OMNIDIRECTIONAL MULTISENSORY PERCEPTION FUSION FOR LONG-TERM PLACE RECOGNITION

This chapter is modified from a conference paper published at the 2018 International Conference on Robotics and Automation (ICRA)[1]. This work was funded in part by the Army Research Office (ARO) grant W911NF-17-1-0447.

Sriram Siva[2], and Hao Zhang[3].

2.1 Abstract

Over the recent years, long-term place recognition has attracted an increasing attention to detect loops for large-scale Simultaneous Localization and Mapping (SLAM) in loopy environments during long-term autonomy. Almost all existing methods are designed to work with traditional cameras with a limited field of view. Recent advances in omnidirectional sensors offer a robot an opportunity to perceive the entire surrounding environment. However, no work has existed thus far to research how omnidirectional sensors can help long-term place recognition, especially when multiple types of omnidirectional sensory data are available. In this chapter, we propose a novel approach to integrate observations obtained from multiple sensors from different viewing angles in the omnidirectional observation in order to perform multi-directional place recognition in long-term autonomy. Our approach also answers two new questions when omnidirectional multisensory data is available for place recognition, including whether it is possible to recognize a place with long-term appearance variations when robots approach it from various directions, and whether observations from various viewing angles are the same informative. To evaluate our approach and hypothesis, we have collected the first large-scale dataset that consists of omnidirectional

[1] ©2018 IEEE. Reprinted with permissions from Sriram Siva, and Hao Zhang. ”Omnidirectional Multisensory Perception Fusion for Long-Term Place Recognition”, in *IEEE International Conference on Robotics and Automation* (ICRA), 2018.

[2] Primary Researcher and Author, Graduate Student, Colorado School of Mines

[3] Associate Professor, Colorado School of Mines

multisensory (intensity and depth) data collected in urban and suburban environments across a year. Experimental results have shown that our approach is able to achieve multi-directional long-term place recognition, and identifies the most discriminative viewing angles from the omnidirectional observation.

2.2 Introduction

Place recognition (or loop closure detection) is an essential component of Simultaneous Localization and Mapping (SLAM), which has been actively studied to achieve SLAM in a loopy environment over the past decades. Recently, place recognition in long-term autonomy has attracted significant attention. Beyond traditional challenges including perceptual aliasing and vision-related issues, long-term place recognition introduces a new, significant challenge – long-term appearance changes [35, 36]. For example, the same outdoor place on a sunny summer noon and during snowy winter evening can look very different. It is recognized [37] that the ability to address long-term appearance variations is essential for robots to perform SLAM during lifelong operations.

Given the importance of long-term place recognition, several representations and matching techniques were proposed mainly to deal with the long-term appearance variation. Both global [38–40] and local [41, 42] features were studied by existing approaches to represent the scene of a place, with an observation that representations based upon global features often perform better [43]. Place matching techniques based on individual frames [44, 45] or frame sequences [35, 46, 47] were also implemented. However, previous long-term visual place recognition methods assumed that observations are acquired from traditional cameras with a limited field of view. Also, all existing methods perform uni-directional place recognition, assuming that a robot goes back to a previously visited place facing the same direction.

In this work, we investigate the problem of long-term place recognition based on omnidirectional perception that allows a robot to perceive its whole surrounding environment. In particular, we are interested in answering two new technical questions, which have not been addressed in the existing research yet. The questions include: (1) when an omnidirectional observation is used, is it possible to perform multi-directional long-term place recognition in situations that an autonomous system approaches the same place from opposite directions; and (2) whether all angles of view the same informative, or are certain angles of view more representative than others in the omnidirectional perception?.

Moreover, we introduce a novel principled approach under the mathematical framework of sparse optimization, which is capable of automatically learning the importance of the viewing angles and integrating omnidirectional perception data to perform multi-directional long-term place recognition. Furthermore, we introduce a multisensory data fusion paradigm under the same framework to integrate heterogeneous visual features that are computed from different types of sensors. Due to our approach's ability to incorporate omnidirectional observations from different sensors, we name our proposed unified method *Fusion of Omnidirectional Multisensory Perception* (FOMP).

The contributions of this chapter are threefold:

- We introduce a new research problem, that is long-term place recognition based upon omnidirectional multisensory perception, and introduce two technical questions that are critical in omnidirectional perception but have not yet been studied in existing long-term place recognition literature.

- We propose the novel FOMP approach, which estimates the importance of viewing angles and learns discriminative features, as well as integrates all the information to construct a discriminative representation for long-term place recognition in situations when robots approach the same location from different directions. We also implement a new optimization solver to solve the formulated problem, which is guaranteed to converge to the local optimal solution theoretically.

- We collect and make available a new large-scale dataset to benchmark methods for multisensory omnidirectional long-term place recognition. Extensive experiments are performed using this new dataset to evaluate FOMP, and to answer the proposed technical questions.

2.3 Related Work

In this section, we review previous approaches to long-term place recognition which can be broadly classified into scene matching for place recognition and long-term place recognition using representations.

2.3.1 Scene Matching for Place Recognition

Many techniques have been implemented to match a query observation and the scene template of a previously visited location. Based on the approach they take towards matching locations, these methods could be broadly classified into two categories. That is, either they use a sequence of images to assert match between two scenes or they follow a image-to-image matching paradigm. Sequence-based matching depends on a sequence of images to find the best matches between query and template image sequences, such as used by SeqSLAM [35] and RAT-SLAM [46]. SeqSLAM computes a summation of similarity scores of query images and template sequences to find the best match location. Typically, methods based on image-to-image matching calculates the distance metric between a query image and existing templates, with the maximum score indicating a scene match [48]. Several techniques also use nearest neighbours search for finding the best match. For example, FAB-MAP [45] uses a Chow-Liu tree to get the best match and RTAB-MAP uses a K-d tree to perform the nearest neighbor search.

The previous methods are designed to work with traditional cameras with limited views, and cannot integrate omnidirectional observations, or perform multi-directional long-term place recognition, which is the focus of our research.

2.3.2 Representation in Long-Term Place Recognition

Most existing place-recognition methods rely on features to construct a representation with the hope to capture long-term scene variations. When environments show changes in illumination conditions, global features outperform local features [35, 37, 43]. Global features extract features from the whole image, and often create a representation using histograms. For example, HOG [39] uses unsigned gradient changes within each pixels of a grid and stores it as a histogram. GIST features [38, 49] employ Gabor filters at different orientations and frequencies to extract information from the images. Convolutional neural networks (CNN) [50–52] are employed to create a representation for matching image sequences. Local Binary Patterns (LBP) are used to encode scenes by labelling pixels of an image by thresholding the neighbourhood of each pixel, which constructs a representation denoted as a binary vector [40]. Depth information from Kinect-like sensors [53] is also been used for object-based SLAM.

Recent methods such as [54] show advantages to identify important features and fuse them together to achieve better performance on long-term place recognition. In this research, we follow the same insight and integrate feature learning as a part of the proposed approach under the unified optimization framework to improve place recognition during long-term autonomy, through fusing heterogeneous features from various omnidirectional sensors.

2.4 The FOMP Approach

We aim at addressing the new problem of utilizing omnidirectional observations to perform long-term place recognition when robots revisit the same place from different directions. To address this challenge, we propose the FOMP approach to estimate the importance of viewing angles and sensor modalities, and to integrate all multisensory omnidirectional data to perform multi-directional long-term place recognition.

Notation. Matrices are denoted using boldface-capital letters, and vectors are denoted by boldface lower-case letters. Given a matrix $\mathbf{U} = \{u_{ij}\} \in \mathbb{R}^{n \times m}$, we denote the i-th row and j-th column as $\mathbf{u}^i$ and $\mathbf{u}_j$ respectively. The ℓ_1-norm of a vector $\mathbf{u} \in \mathbb{R}^n$ is defined as $\|\mathbf{u}\|_1 = \sum_{i=1}^n |u_i|$. The ℓ_2-norm of a vector $\mathbf{u}$ is defined as $\|\mathbf{u}\|_2 = \sqrt{\mathbf{u}^\top \mathbf{u}}$. The Frobenius norm of a matrix $\mathbf{U}$ is defined as $\|\mathbf{U}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n u_{ij}^2}$.

2.4.1 Problem Formulation

Given a collection of omnidirectional images acquired in different scenarios, each image is equally divided into a set of views. We also assume that multimodal features are extracted from each view, where a modality of features is defined as the features computed using a specific descriptor from images acquired by a specific visual sensor (e.g., intensity or depth sensor). Then, the set of multisensory omnidirectional images can be expressed as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$, where $\mathbf{x}_i \in \mathbb{R}^p$ denotes the feature vector extracted from all views of the i-th image, which is a concatenation of features from m modalities, such that $p = \sum_{i=1}^{m} \sum_{j=1}^{a} d_{ij}$, where d_{ij} is the dimensionality of the i-th modality in the j-th view, and a is the total number of views. The label vector of scenarios (e.g., different seasons) associated with $\mathbf{X}$ is represented by $\mathbf{Y} = [\mathbf{y}_1; \mathbf{y}_2; \ldots; \mathbf{y}_n] \in \mathbb{R}^{n \times c}$, where c is the number of scenarios, and $\mathbf{y}_i$ is the scenario indicating vector, with elements $y_{ij} \in \{0, 1\}$ representing that the i-th image is collected from the j-th scenario.

Then, the problem of omnidirectional multisensory place recognition is formulated as a regularized sparse optimization problem:

$$\min_{\mathbf{W}} \mathcal{L}(\mathbf{X}, \mathbf{Y}; \mathbf{W}) + \lambda \mathcal{R}(\mathbf{W}) \tag{2.1}$$

where $\mathcal{L}(\cdot)$ is a loss function and $\mathcal{R}(\cdot)$ is a sparsity-inducing regularizer with $\lambda \geq 0$ as a trade-off hyperparameter. $\mathbf{W}$ denotes the weight matrix, which represents the importance of the features $\mathbf{X}$ to represent the scenarios $\mathbf{Y}$ in general.

We define a new loss function to address multi-directional place recognition (i.e., identification of the same place from different directions) as follows:

$$\min_{\mathbf{W}} \|(\mathbf{R}\mathbf{X})^T \mathbf{W} - \mathbf{Y}\|_F^2 \tag{2.2}$$

where $\mathbf{R}$ is the rotation matrix for aligning omnidirectional images with the same origin. For example, when two omnidirectional images are take by a car driving on the two sides of a road respectively, then one image must be rotated $180°$ to align with the other image, which is performed using $\mathbf{R}$.

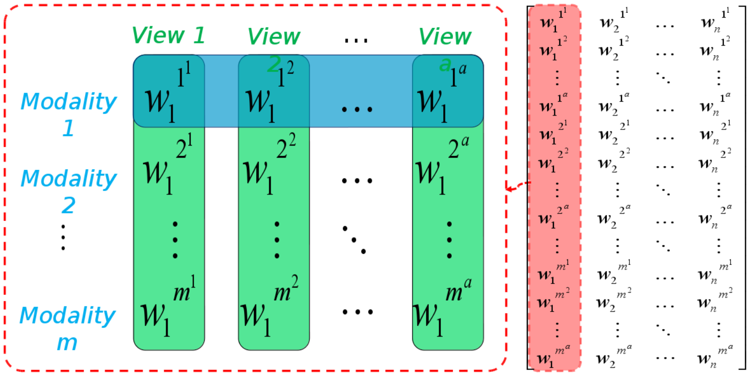

View 1
View 2
View a
Modality 1
Modality 2
Modality m
$w_1^{1^1}$ $w_1^{1^2}$ $w_1^{1^a}$
$w_1^{2^1}$ $w_1^{2^2}$ $w_1^{2^a}$
$w_1^{m^1}$ $w_1^{m^2}$ $w_1^{m^a}$
$w_2^{1^1}$ … $w_n^{1^1}$
$w_n^{m^a}$

Figure 2.1 Illustration of the weight matrix $\mathbf{W}$.

The solution to this optimization task is the weight matrix $\mathbf{W} = [\mathbf{w_1}, \mathbf{w_2}, \dots, \mathbf{w_c}] \in \mathbb{R}^{p \times c}$, which contains the weights $\mathbf{w}_i \in \mathbb{R}^p$ of all modalities and views with respect to the i-th scenario. Each $\mathbf{w}_i$ contains weights of m-modalities from all views, which can be expanded as $\mathbf{w}_i = [\mathbf{w}_i^1, \mathbf{w}_i^2, \dots, \mathbf{w}_i^m]^T$. In addition, the weight vector of each modality $\mathbf{w}_i^m$ can be further divided into a segments, each from a particular view, as follows $\mathbf{w}_i^m = [\mathbf{w}_i^{m^1}; \mathbf{w}_i^{m^2}; \dots; \mathbf{w}_i^{m^a}] \in \mathbb{R}^{d_{ij}}$, representing the weights of the features extracted in different views from m-th modality and i-th scenario. The weight matrix is graphically represented in Figure 2.1.

2.4.2 Learning Discriminative Views

An omnidirectional image provides $360°$ field of view, that allows robots to observe the entire surrounding environment. We hypothesize that for long-term place recognition, specific views in the omnidirectional image are more discriminative than others. In order to automatically identify these views, we introduce a novel cone-structured sparsity-inducing norm to learn the discriminative features under our unified regularized optimization framework.

Formally, each leaf node of the introduced cone structure contains the features extracted from an individual view, and each internal node of the cone contains the features from its respective child nodes, which represents a combination of the views represented by the child nodes. We represent the set of nodes as $\mathcal{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{root})$, each $\mathbf{v}$ containing features collected from a certain view or multiple adjacent views, and we denote the weights of the features from respective views as $(\mathbf{w}_{\mathbf{v}_1}, \mathbf{w}_{\mathbf{v}_2}, \ldots, \mathbf{w}_{\mathbf{v}_{root}})$. For example, the cone structure with four leaf nodes, each including the features obtained from a 90° view, is demonstrated in Figure 2.2, with the 3D cone structure illustrated in Figure 2.2(a) and the unwrapped structure shown in Figure 2.2(b).

Then, we propose a method to compute the *weight of each node* $w(\mathbf{v})$ in the cone structure, which is a scalar that indicates the importance of the node, as follows:

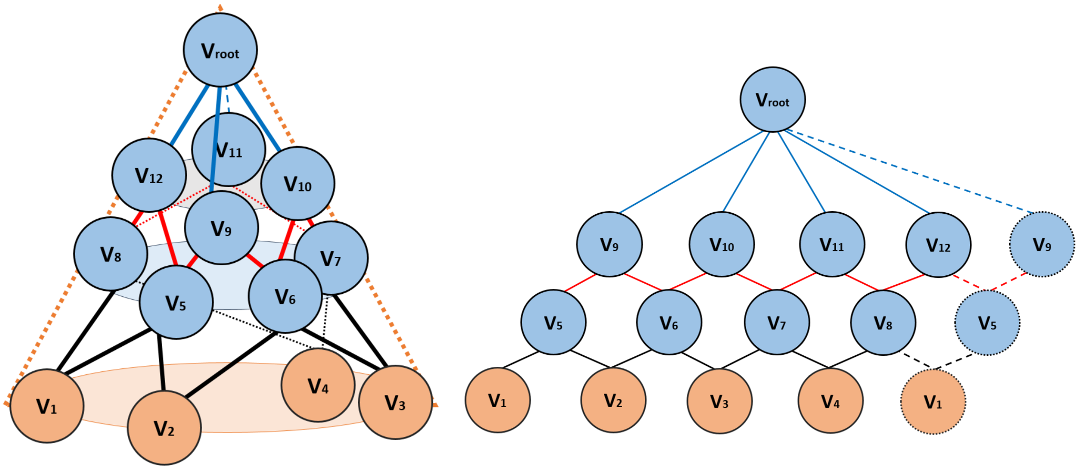
$$w(\mathbf{v}) = \begin{cases} \widetilde{h}_{\mathbf{v}} \sum_{C(\mathbf{v})} \|\mathbf{w}_{C(\mathbf{v})}\|_1 + h_{\mathbf{v}} \sum_{C(\mathbf{v})} \|\mathbf{w}_{C(\mathbf{v})}\|_2 \\ \qquad\qquad\qquad\qquad\qquad \text{if } \mathbf{v} \text{ is an internal node} \\ \sum_{\mathbf{v}} \|\mathbf{w}_{\mathbf{v}}\|_1 \qquad\qquad \text{if } \mathbf{v} \text{ is a leaf node} \end{cases}$$

where $\widetilde{h}_{\mathbf{v}} = 1 - h_{\mathbf{v}}$, and $h_{\mathbf{v}}$ is the normalized height of a node $\mathbf{v}$ with respect to the height of the cone structure. At lower levels of the cone structure, $h_{\mathbf{v}}$ takes smaller values; thus ℓ_2-norm is more significant. Moving toward the upper level of the cone, $h_{\mathbf{v}}$ increases, so ℓ_1-norm becomes more dominant. The variable $h_{\mathbf{v}}$ is designed to incorporate the principle that at lower levels, the grouping effect of different views should be promoted and at higher levels sparsity among the grouped views should be emphasized.

Based upon the weights of all nodes, the cone-structured sparsity-inducing norm is defined as $\|\mathbf{W}\|_C = \sum_{\mathbf{v}\in\mathcal{V}} w(\mathbf{v})$. which allows for discovering discriminative views by assigning a greater weight to features from discriminative views. When using the cone-structure sparsity-inducing norm as a regularization term, we obtain the new objective function:

$$\min_{\mathbf{W}} \|(\mathbf{RX})^T\mathbf{W} - \mathbf{Y}\|_F^2 + \lambda\|\mathbf{W}\|_C \tag{2.3}$$

where λ is a hyperparameter used to balance the loss function and the regularization term.



(a) 3D cone structure

(b) Unwrapped structure

Figure 2.2 Illustration of the proposed cone-structured sparsity-inducing norm for learning discriminative views. Figure 2.2(a) illustrates the 3D structure of our proposed cone-structured sparsity-inducing norm and Figure 2.2(b) illustrates the corresponding unwrapped structure of the same sparsity-inducing norm.

2.4.3 Learning Discriminative Modalities

Modern robots are usually equipped with different types of sensors (e.g., intensity and depth sensors), and we can extract different type of features from observations obtained by each sensor. We employ the term *modality* to refer to a set of features extracted by a type of feature extraction method from the data obtained by a specific sensor. In this case, some feature modalities are typically more descriptive than others. For example, in a dark environment, observations from depth sensors are often more useful than data from color cameras. In this research, we also propose to identify discriminative features under the unified optimization framework to improve place recognition accuracy.

Inspired by [54], we incorporate a modality norm, named M-norm, as a regularization term to enforce sparsity among different modalities thus identifying discriminative modalities. The M-norm applies the ℓ_2-norm within each modality and the ℓ_1-norm across different modalities. The M-norm is mathematically expressed as $\|\mathbf{W}\|_M = \sum_{i=1}^{c} \sum_{j=1}^{m} \|\mathbf{w}_i^j\|_2$.

Incorporating both sparsity-inducing norms to model the relationship among various views and modalities, we observe the final objective function:

$$\min_{\mathbf{W}} \|(\mathbf{RX})^T\mathbf{W} - \mathbf{Y}\|_F^2 + \lambda_1\|\mathbf{W}\|_C + \lambda_2\|\mathbf{W}\|_M \tag{2.4}$$

where λ_1 and λ_2 are the trade-off hyper-parameters to balance the loss function and the sparsity-inducing norms.

2.4.4 Omnidirectional Multisensory Place Recognition

After solving the regularized optimization problem in Eq. 2.4 (using Algorithm 1, described in the next subsection), we obtain the optimal weight matrix $\mathbf{W}^* \in \mathbb{R}^{p\times c}$. Given a feature vector $\mathbf{x}_i \in \mathbb{R}^p$ of a query omnidirectional multisensory observation, we can compute a similarity score between this query observation and the template as follows:

$$s = \sum_{i=1}^{a}\sum_{j=1}^{m} w_A(i) * w_M(j) * s^{j^i} \tag{2.5}$$

where s^{j^i} denotes the similarity score between the observation and the template in i-th view of j-th modality, $w_M(j)$ is the optimal weight of the j-th modality, and $w_A(i)$ is the optimal weight of the i-th view. If the similarity score thus calculated is above a user-defined threshold, the query image is decided as matching to the template. The weight of the i-th view is computed as $w_A(i) = \sum_{j=1}^{m} \|\mathbf{w}^{j^i}\|_2, i = 1, 2, \ldots, a$, and the weight of the j-th modality is computed as $w_M(j) = \|\mathbf{w}^j\|_2, j = 1, 2, \ldots, m$.

2.4.5 Optimization Algorithm

The objective function in Eq. (2.4) comprises of non-smooth regularization terms, which is challenging to solve in general. Thus, we implement a new iterative algorithm, that follows the iterative re-weighted method [55], to solve this formulated optimization problem.

Algorithm 1: An iterative algorithm to solve the formulated optimization problem in Eq. (4)

```
Input  : feature matrix $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{p \times n}$ and ground truth matrix
         $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_n]^T \in \mathbb{R}^{n \times c}$ from the training set
Let $t = 1$. Initialize $\mathbf{W}(t)$ by solving $\min_{\mathbf{W}} \|(\mathbf{RX})^T\mathbf{W} - \mathbf{Y}\|_F^2$.
while not converge do
    Calculate the block diagonal matrix $\mathbf{D}(t+1)$, where the $k$-th diagonal block of
      $\mathbf{D}(t+1)$ is $\frac{1}{2\|\mathbf{w}^k(t)\|_2}$.
    Calculate the block diagonal matrix $\widetilde{\mathbf{D}}(t+1)$, where the diagonal block of $\widetilde{\mathbf{D}}(t+1)$ is
      $\frac{1}{2\|\mathbf{W}(t)\|_C}\mathbf{I}_{id}$.
    For each $\mathbf{w}_i (1 \le i \le c)$,
      $\mathbf{w}_i(t+1) = ((\mathbf{RX})(\mathbf{RX})^T + \gamma_1\mathbf{D}(t+1) + \gamma_2\widetilde{\mathbf{D}}(t+1))^{-1}(\mathbf{RX})\mathbf{y}_i$.
    $t = t + 1$.
Output : $\mathbf{W} = \mathbf{W}(t) \in \mathbb{R}^{p \times c}$
```

Taking the derivative of the objective function with respect to the columns of $\mathbf{W}$ (i.e., $\mathbf{w}_i, i = 1, \ldots, c$) and setting the whole equation to a zero vector gives us:

$$(\mathbf{RX})(\mathbf{X}^T\mathbf{R}^T)\mathbf{w}_i - (\mathbf{RX})\mathbf{y}_i + \gamma_1\mathbf{D}\mathbf{w}_i + \gamma_2\widetilde{\mathbf{D}}\mathbf{w}_i = 0 \tag{2.6}$$

where $\widetilde{\mathbf{D}}$ is a diagonal matrix with the i^{th} diagonal element as $\dfrac{1}{2\|\mathbf{w}^i\|_2}$, and $\mathbf{D}$ is defined as the diagonal matrix with the diagonal block as $\dfrac{1}{2\|\mathbf{W}\|_C}\mathbf{I}_{id}$, where $\mathbf{I}_{id}$ is an identity matrix of size p. Since the matrices $\mathbf{D}$ and $\widetilde{\mathbf{D}}$ are dependent on the vectors of $\mathbf{W}$, we develop an iterative algorithm to solve the optimization problem with these unknown variables, as described in Algorithm 1, which holds a theoretical convergence guarantee as described by the following theorem.

Theorem 1: Algorithm 1 converges to the local optimal solution to the optimization problem in Eq. 2.4.

Proof. In the following, we prove that Algorithm 1 decreases the value of the objective function with each iteration and converges to the local optimal value. But first, we present a lemma:

Lemma 1: For any two given vectors $\mathbf{v}$ and $\tilde{\mathbf{v}}$, the following inequality relation holds:

$$\|\tilde{\mathbf{v}}\|_2 - \frac{\|\tilde{\mathbf{v}}\|_2^2}{2\|\mathbf{v}\|_2} \le \|\mathbf{v}\|_2 - \frac{\|\mathbf{v}\|_2^2}{2\|\mathbf{v}\|_2}$$

Proof. We have:

$$-(\|\widetilde{\mathbf{v}}\|_2 - \|\mathbf{v}\|_2)^2 \leq 0$$

$$-\|\widetilde{\mathbf{v}}\|_2^2 - \|\mathbf{v}\|_2^2 + 2\|\widetilde{\mathbf{v}}\|_2\|\mathbf{v}\|_2 \leq 0$$

$$2\|\widetilde{\mathbf{v}}\|_2\|\mathbf{v}\|_2 - \|\widetilde{\mathbf{v}}\|_2^2 \leq \|\mathbf{v}\|_2^2$$

$$\|\widetilde{\mathbf{v}}\|_2 - \frac{\|\widetilde{\mathbf{v}}\|_2^2}{2\|\mathbf{v}\|_2} \leq \|\mathbf{v}\|_2 - \frac{\|\mathbf{v}\|_2^2}{2\|\mathbf{v}\|_2}$$

□

From Lemma 6, we can derive the following corollary:

Corollary 1: For any two given matrices $\mathbf{M}$ and $\tilde{\mathbf{M}}$, the following inequality relation holds:

$$\|\tilde{\mathbf{M}}\|_C - \frac{\|\tilde{\mathbf{M}}\|_C^2}{2\|\mathbf{M}\|_C} \leq \|\mathbf{M}\|_C - \frac{\|\mathbf{M}\|_C^2}{2\|\mathbf{M}\|_C}$$

According to Steps 3 of Algorithm 1, we calculate $\mathbf{W}(t+1)$ as:

$$\mathbf{W}(t+1) = \min_{\mathbf{W}} \|\mathbf{R}\mathbf{X}^\top\mathbf{W} - \mathbf{Y}\|_F^2 + \gamma_1 Tr\mathbf{W}^\top\mathbf{D}(t+1)\mathbf{W} + \gamma_2 Tr\mathbf{W}^\top\tilde{\mathbf{D}}(t+1)\mathbf{W} \tag{2.7}$$

Then, we can derive that

$$\begin{aligned}\mathcal{J}(t+1) + \gamma_1 Tr\mathbf{W}^\top(t+1)\mathbf{D}(t+1)\mathbf{W}(t+1) + \gamma_2 Tr\mathbf{W}^\top(t+1)\tilde{\mathbf{D}}(t+1)\mathbf{W}(t+1) \\ \leq \mathcal{J}(t) + \gamma_1 Tr\mathbf{W}^\top(t)\mathbf{D}(t)\mathbf{W}(t) + \gamma_2 Tr\mathbf{W}^\top(t)\tilde{\mathbf{D}}(t)\mathbf{W}(t)\end{aligned} \tag{2.8}$$

where $\mathcal{J}(t) = \|\mathbf{R}\mathbf{X}^\top\mathbf{W}(t) - \mathbf{Y}\|_F^2$.

After substituting the definition of $\mathbf{D}$ and $\tilde{\mathbf{D}}$, we obtain

$$\begin{aligned} &\mathcal{J}(t+1) + \gamma_1 \frac{\|\mathbf{W}(t+1)\|_C^2}{2\|\mathbf{W}(t)\|_C} + \gamma_2 \textstyle\sum_{i=1}^m \frac{\|\mathbf{w}^i(t+1)\|_2^2}{2\|\mathbf{w}^i(t)\|_2} \\ \leq \quad &\mathcal{J}(t) + \gamma_1 \frac{\|\mathbf{W}(t)\|_C^2}{2\|\mathbf{W}(t)\|_C} + \gamma_2 \textstyle\sum_{i=1}^m \frac{\|\mathbf{w}_i(t)\|_2^2}{2\|\mathbf{w}^i(t)\|_2} \end{aligned} \tag{2.9}$$

From Lemma 6 and Corollary 4, we can derive that

$$\|\mathbf{W}(t+1)\|_C - \frac{\|\mathbf{W}(t+1)\|_C^2}{2\|\mathbf{W}(t)\|_C} \quad \leq \quad \|\mathbf{W}(t)\|_C - \frac{\|\mathbf{W}(t)\|_C^2}{2\|\mathbf{W}(t)\|_C} \tag{2.10}$$

and,

$$\sum_{i=1}^m \|\mathbf{w}^i(t+1)\|_2 - \sum_{i=1}^m \frac{\|\mathbf{w}^i(t+1)\|_2^2}{2\|\mathbf{w}^i(t)\|_2} \quad \leq \quad \sum_{i=1}^m \|\mathbf{w}^i(t)\|_2 - \sum_{i=1}^m \frac{\|\mathbf{w}^i(t)\|_2^2}{2\|\mathbf{w}^i(t)\|_2} \tag{2.11}$$

Adding Eq. (7.14)-(4.16) on both sides, we have

$$\begin{aligned} &\mathcal{J}(t+1) + \gamma_1 \|\mathbf{W}(t+1)\|_C + \gamma_2 \textstyle\sum_{i=1}^m \|\mathbf{w}^i(t+1)\|_2 \\ &\leq \mathcal{J}(t) + \gamma_1 \|\mathbf{W}(t)\|_C + \gamma_2 \textstyle\sum_{i=1}^m \|\mathbf{w}^i(t)\|_2 \end{aligned} \tag{2.12}$$

Eq. 7.16 decreases the value of the objective function with each iteration. As our objective function is convex, Algorithm 6 converges to the local optimal value. Therefore, Algorithm 1 converges to the local optimal solution to the optimization problem in Eq. (4). □

2.5 Omnidirectional Multisensory Dataset

One of the contributions of this research is the collection of large-scale omnidirectional multisensory datasets. Although various sensors and omnidirectional cameras are increasingly widely deployed on robots and autonomous cars, before this research, no dataset containing omnidirectional multisensory information is publicly available for benchmarking long-term place recognition. Motivated by this need, we collected the new dataset called MOLP, which stands for *Multimodal Omnidirectional Long-term Place-recognition*. The dataset was collected using a omnidirectional camera installed on a SUV to collect omnidirectional intensity and depth information. The MOLP dataset includes two sub-datasets obtained from two different routes:

- Route-A: *Mines-Downtown Golden*. This route contains scenes from the Colorado School of Mines and downtown of Golden CO. This route is 4.3 miles long and the dataset consists of 3000-7000 images in each of the 16 instances from different long-term scenarios across a year. The dataset also captures the short-term dynamics such as traffics, construction work, and pedestrians.

- Route-B: *Historic Suburban Golden*. This route contains scenes of the trip where the gold-rush era started 150 year ago. This driving route is 7.6 miles long, from the circuitous suburban Golden and to the Rocky Mountains. The dataset consists of 2500-5000 images in each of the 16 instances from different long-term scenarios across a year. Beyond long-term appearance changes, this route has the severe challenge of perceptual aliasing because of the similar winding roads while driving.

The MOLP dataset is publicly available and more details are discussed on the dataset website at http://hcr.mines.edu/code/MOLP.html.

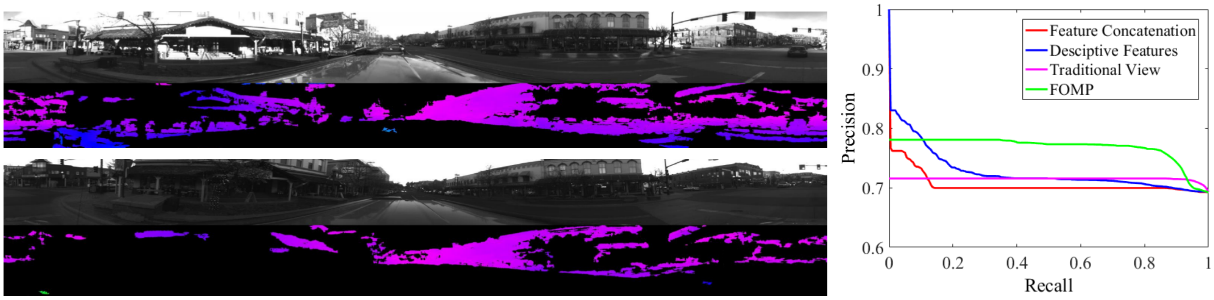
2.6 Experiments

To evaluate the FOMP method on long-term place recognition based on multisensory omnidirectional observations, we performed extensive experiments using the newly collected MOLP dataset.

2.6.1 Experimental Setup

The Summer and Fall scenarios from the dataset are used in the experiments. Each omnidirectional image is vertically split into 18 sections, which corresponds to 18 views, each including 20° field of view. Then, each split image is down-sampled to a resolution of 210*240, which consists of both intensity and depth information. No image processing is further performed on these images. For ground truth we use the GPS data recorded at the time of collecting the dataset.

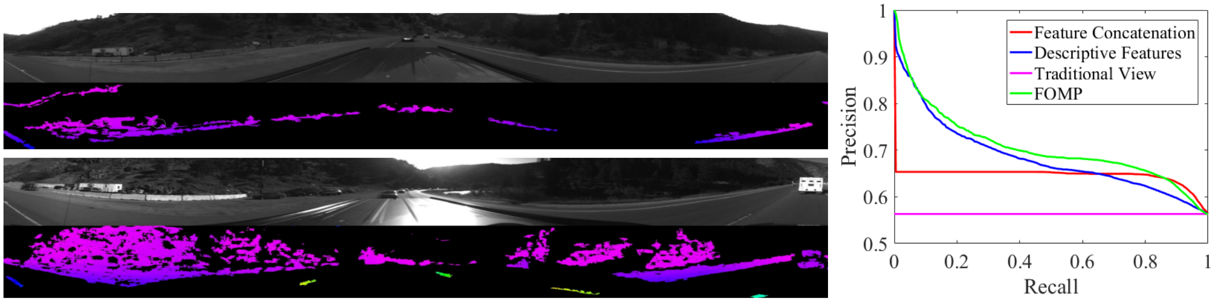
Four different types of visual features are extracted from each of the intensity and depth images in our experiments, including GIST [38], HOG [39], LBP [40], and CNN-based deep features [50, 51]. All features are separately extracted from both intensity and depth images. Moreover, we implement several techniques from the literature to compare with our FOMP approach, including representations by concatenated features and discriminative features. Furthermore, we implement a method using the traditional front $80°$ field of view as a baseline. For all experiments, we set 0.1 to the hyper-parameter λ_1, and 0.05 to λ_2.



(a) Matched omnidirectional multisensory observations

(b) Precision-recall curves

Figure 2.3 Results over Route-A across different times of a day (i.e., morning versus evening).



(a) Matched omnidirectional multisensory observations

(b) Precision-recall curves

Figure 2.4 Results over Route-B across different times of a day (i.e., morning versus evening).

2.6.2 Results at Different Times of a Day

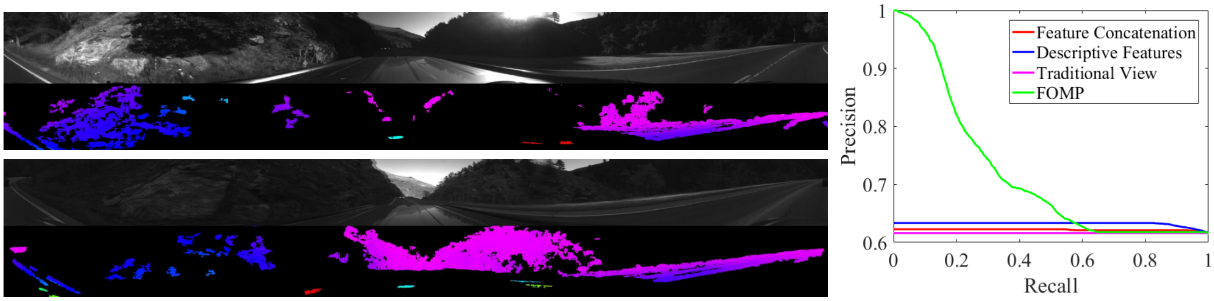
To evaluate our approach on place recognition with long-term appearance changes across different times of a day and with environment dynamics, we perform experiments using morning and evening scenarios for both Route-A and Route-B in the MOLP dataset.

The qualitative result on the Route-A dataset is illustrated in Figure 2.3(a), which includes a detected match of omnidirectional intensity-depth images in the query and template. It can be observed from the intensity image that the same place exhibits very different illumination conditions in the morning versus evening, and also contains different dynamics because of varying traffic and pedestrians. In this challenging scenario, the match shows our FOMP approach can well perform place recognition with long-term appearance variations.

In the experiments over the Route-B dataset, we observe similar qualitative results, with an exemplary match depicted in Figure 2.7, which shows our FOMP approach is able to match the places with the presence of long-term illumination changes.

To provide a quantitative evaluation, the standard metric of precision-recall curves is used. Figure 2.3(b) shows the precision-recall curves obtained over the morning and evening scenes of Route-A. The results for Route-B is illustrated in Figure 2.7(b). Comparisons with baseline techniques are also presented in the respective figures. We observe that integrating omnidirectional information can improve the performance (as shown in Figure 2.3(b)). In addition, our FOMP method outperforms the baseline techniques, due to the ability to identify and fuse discriminative views and feature modalities.

In addition, we perform experiments to evaluate the importance of different viewing angles, with the results shown in Figure 2.6(a) for Route-A. The relative importance of each view is presented as a heat chart, with a warmer color denoting greater importance, and the vehicle is facing up as the front. It can be observed that the front and back views in the omnidirectional observation are the most descriptive for long-term place recognition across different times of the day, and the sideways of the observation are less descriptive. Finally, we perform experiments to assess the importance of different sensing modalities, i.e., different types of features acquired from various sensors. The results are shown in Figure 2.6(b) for Route-A, with the numbers denoting the importance of the modality. We observe consistent results on modality importance: the four most important modalities are depth-LBP, depth-GIST, intensity-GIST, and depth-CNN.



(a) Matched omnidirectional multisensory observations

(b) Precision-recall curves

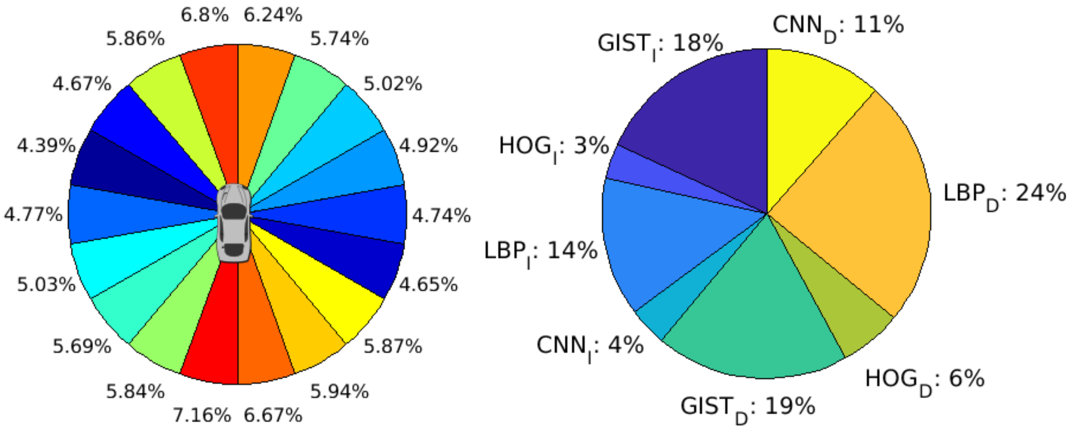
Figure 2.5 Results over the Route-B dataset across different seasons (i.e., Summer versus Fall).

2.6.3 Results across Different Seasons

To evaluate FOMP over a longer span of time, we perform experiments using omnidirectional multisensory observations across different seasons, in which the places show significant appearance changes caused by weather and vegetation.

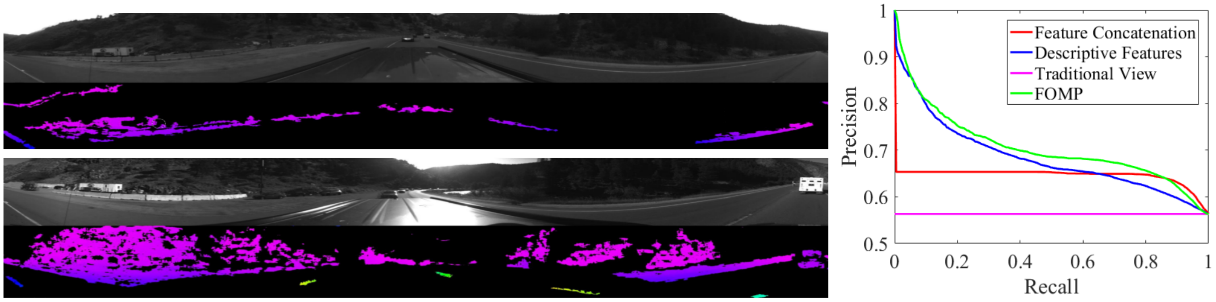
As the qualitative experimental result, examples of the detected omnidirectional multisensory matches between query and template observations are illustrated in Figure 2.5(a) for Route-B. The results show FOMP is able to well perform place recognition under different vegetation and illumination conditions across different seasons. The precision-recall curves are illustrated in Figure 2.5(b). We can observe that the proposed approach outperforms the baseline techniques on long-term place recognition across different seasons.

The importance of viewing angles for Route-B across different seasons is illustrated in Figure 2.8(a). Results in Route-B are a bit different from previous observations: the front view is relatively more discriminative than the back view, although both front and back views are still more important than side views for omnidirectional long-term place recognition. Finally, the quantitative results over modality weights are illustrated in Figure 2.8(b) for Route-B across different seasons, respectively. Similar to the results obtained from different times of a day, the modality importance is also consistent with the same top four most descriptive modalities. When we sum up the weights of features acquired by either the intensity or depth sensor, we observe depth data weights more over intensity data, which indicates that depth observations are more important. A possible explanation is that the intensity cues are more sensitive to the long-term appearance variation such as various illumination and weather, while the depth information determined by the environment topology is less affected by appearance changes.



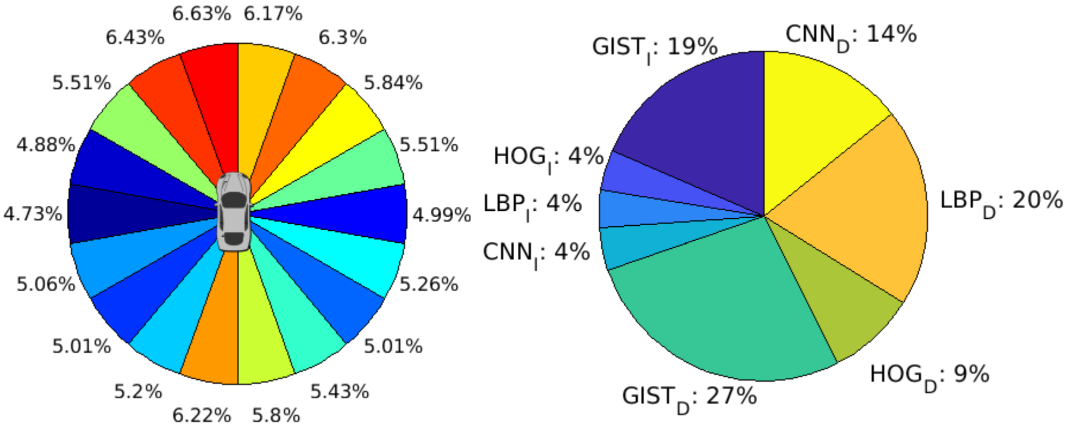
(a) Descriptive views (b) Descriptive modalities

Figure 2.6 Experimental results on view and modality importance using the Route-A dataset across various times of a day.



(a) Matched omnidirectional multisensory observations (b) Precision-recall curves

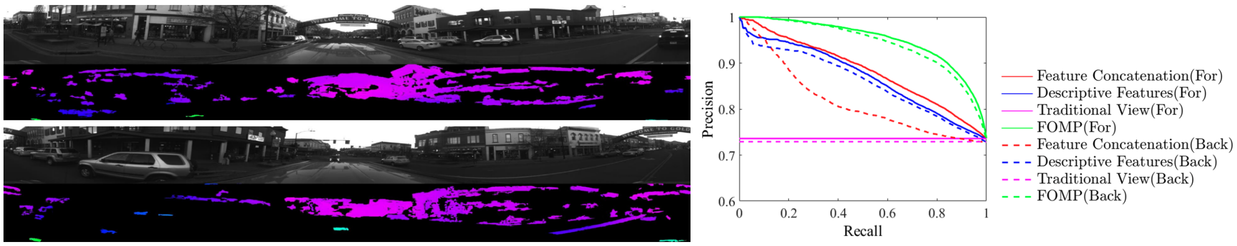
Figure 2.7 Results over the Route-B dataset across different times of a day (i.e., morning versus evening).



(a) Descriptive Views

(b) Descriptive Modalities

Figure 2.8 Experimental results on view and modality importance using the Route-B dataset across different seasons.



(a) Detected matches when the car drives from opposite directions

(b) Precision-recall curves

Figure 2.9 Results of bidirectional place recognition when a car drives through the same place from different directions in different seasons. In the legend of precision-recall curves, "*For*" indicates the FOMP approach is trained using forward-direction data only, and "*Back*" indicates the approach is trained using backward-direction data only. Testing is performed using data collected from both directions.

2.6.4 Results on Multi-directional Place Recognition

One unique capability of the FOMP approach is to achieve multi-directional place recognition, which is enabled by introducing the rotation matrix in our problem formulation to address different robot's orientations. This new advantage is validated and evaluated in this set of experiments. We train our approach using the data from one direction only (either forward or backward), then we evaluate its performance on long-term place recognition using data from both directions.

An exemplary location match detected by FOMP is illustrated in Figure 2.9(a). Because the vehicle drives through the same place from different directions, it is observed that the query observation is rotated around 180° comparing to the template, which is clearly demonstrated by the arched sign (which reads "WELCOME TO GOLDEN"). Since the rotation matrix $\mathbf{R}$ in our formulation is able to model this rotation, FOMP can successfully recognize the same place when the car approaches it from different directions. Precision-recall curves obtained from bidirectional place recognition are illustrated in Figure 2.9(b). We can observe three key phenomena. First, for bidirectional place recognition, the methods using traditional camera do not work, as expected. Second, our FOMP approach significantly outperforms other baseline techniques, mainly due to its capability of modeling rotations. Third, because of the same reason, FOMP methods trained on forward-direction data or backward-direction data obtain consistent good performance. This consistent capability highlights our approach for bidirectional place recognition in long-term autonomy.

2.7 Summary

In this chapter, we propose a new problem of place recognition from omnidirectional multisensory observations in long-term autonomy. To address this challenge, we introduce the novel FOMP approach that is able to identify and integrate discriminative multimodal data obtained from heterogeneous sensors in different views. Our approach shows that different viewing angles in the omnidirectional observation have different description powers. Our research also demonstrates that multi-directional long-term place recognition is achievable. To validate our FOMP approach and hypothesis, we collect a large-scale dataset containing omnidirectional multisensory observations. Experimental results on this dataset have demonstrated that FOMP obtains promising long-term place recognition performance.

CHAPTER 3
ROBOT PERCEPTUAL ADAPTATION TO ENVIRONMENT CHANGES FOR LONG-TERM HUMAN TEAMMATE FOLLOWING

This chapter is modified from a joural paper published at the 2020 International Journal on Robotics Research (IJRR)[4]. This research was partially supported by Army Research Office (ARO) under Grant W911NF-17-1-0447, U.S. Air Force Academy (USAFA) under Grant FA7000-18-2-0016, and the Distributed and Collaborative Intelligent Systems and Technology (DCIST) CRA under Grant W911NF-17-2-0181.

Sriram Siva[5], and Hao Zhang[6].

3.1 Abstract

Perception is one of the several fundamental abilities required by robots, and it also poses significant challenges, especially in real-world field applications. Long-term autonomy introduces additional difficulties to robot perception, including short and long-term changes of the robot operation environment (e.g., lighting changes). In this paper, we propose an innovative human-inspired approach named *RObot Perceptual Adaptation* (ROPA) that is able to calibrate perception according to the environment context, which enables perceptual adaptation in response to environmental variations. ROPA jointly performs feature learning, sensor fusion, and perception calibration under a unified regularized optimization framework. We also implement a new algorithm to solve the formulated optimization problem, which has a theoretical guarantee to converge to an optimal solution. In addition, we collect a large-scale dataset from physical robots in the field, called *Perceptual Adaptation to Environment Changes* (PEAC), with the aim to benchmark methods for robot adaptation to short-term and long-term, and fast and gradual lighting

[4] ©2020 Sage Journals. Reprinted with permissions from Sriram Siva, and Hao Zhang. ”Robot Perceptual Adaptation to Environment Changes for Long-Term Human Teammate Following”, in *The International Journal of Robotics Research* (IJRR), 2020.

[5] Primary researcher and author, Graduate Student, Colorado School of Mines

[6] Associate Professor, Colorado School of Mines

changes for human detection based upon different feature modalities extracted from color and depth sensors. Utilizing the PEAC dataset, we conduct extensive experiments in the application of human recognition and following in various scenarios to evaluate ROPA. Experimental results have validated that the ROPA approach obtains promising performance in terms of accuracy and efficiency, and effectively adapts robot perception to address short-term and long-term lighting changes in human detection and following applications.

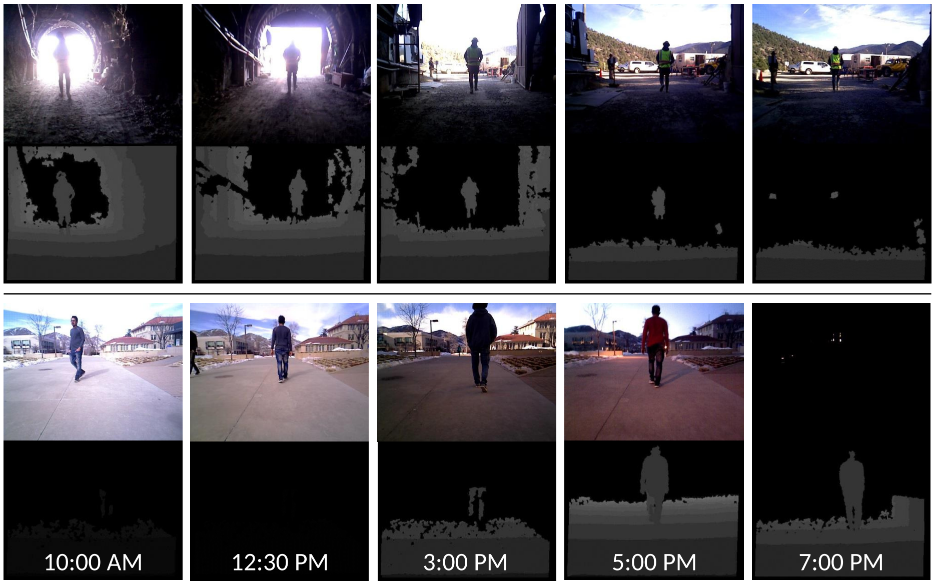


Figure 3.1 Motivating examples of robot perceptual adaptation in human teammate following applications. When a robot follows a human during a long-term operation (e.g., search and rescue), the robot requires the capability of perceptual adaptation to adapt to fast changes (e.g., when moving from a dark tunnel to a bright open area shown in the top row) and long-term changes (e.g., different times of the day shown in the bottom row) and in order to avoid perception failures. The problem of robot adaptation to fast and dramatic environment changes has not yet been well addressed. For example, given a stream of color and depth data as the input, existing methods of human detection cannot adaptively choose sensing modalities and often lose the person when the robot travels from a dark mine to a bright open area with fast and dramatic lighting changes.

3.2 Introduction

Perception is an essential capability for autonomous robots to perceive the surrounding world and their own states so that they can accomplish other fundamental functionalities, such as navigation and human-robot teaming. For example, robots following humans to perform search and rescue missions and robots working collaboratively in human-robot teaming both need the ability to collect information about their environments to make decisions; robots also need to collect information about themselves and terrain characteristics to perform navigation in unstructured field environments.

While the robot perception research community has made impressive strides over the past years, it remains an unsolved problem for real-world field robotics applications, especially for robots that operate in dynamic and unstructured field environments. Several key challenges in robot perception must be addressed. As modern robotic platforms are often equipped with a variety of sensors it is necessary to effectively integrate this multisensory data. In addition, there is often limited on-board computing power for robots operating in field environments, and efficient algorithms are required to deliver real-time performance with this limited computational power.

In the past several years, the robotics community has paid more attention towards long-term autonomy — robots that are able to operate for days, months, years, and eventually a lifetime. Long-term autonomy introduces new additional challenges to robot perception. For example, as demonstrated in Figure 3.1, when robots follow a human teammate to perform search and rescue missions over long periods of time, robots often need to operate in indoor and outdoor environments, navigate between dark tunnels and bright open areas, and perform human detection under significant illumination variations at different time of the day. These changing, unstructured field environments often cause failures for robot perception, such as an inability to continue tracking teammates. Thus, addressing the short-term and long-term environment changes is vital to enable long-term autonomy.

Several methods were previously implemented to address the problem of long-term autonomy in dynamic unstructured environments. A widely used paradigm is to learn a unified representation of unstructured environments, which can be applied to various scenarios at different time. For example, in long-term place recognition (a.k.a. loop closure detection), methods based on holistic layouts [54, 56] or landmarks [50, 57] were designed to construct a representation of the environment that is robust to long-term variations over time to find a match with previously visited places. These techniques look past the changes of the environments over discrete time points (e.g. morning vs evening and summer vs winter) to determine the underlying place features that are most representative. However, these techniques learn fixed unified representations that do not adapt or change according to environment dynamics. A conventional robot adaptation paradigm is based upon case-based reasoning [58], which accomplishes adaptation by switching between multiple perception modalities depending upon the current context. However, they are limited to cases that have been manually predefined, which makes them impractical for dynamically changing unstructured environments that may have a large number of context cases in a continuous and high-dimensional space. In addition, online learning [59], [60], [61] was also widely studied, where adaptation is achieved by continuously training the model using streams of data in an online fashion. Because online learning models drift and often require many iterations to converge to an optimal model again, they are less effective in scenarios when a robot needs to adapt to fast or repeated changes of the environment.

In this paper, we propose a novel approach named *RObot Perceptual Adaptation* (ROPA) that learns a dynamical fusion of multisensory perception data, which is adaptive to continuous short-term and long-term environment changes. ROPA is inspired by the observation that the human eye is able to adapt to a wide range of lighting conditions, and by the phycological findings in perceptual adaptation of humans. Human perceptual adaptation is a fundamental property of perceptual processing to “calibrate perception to current inputs” [62] and to “maintain the match between visual coding and the visual environment” [63, 64]. Our application in this paper focuses on human detection based upon different types of features extracted from color and depth sensors installed on a mobile robot to perform long-term human teammate following. During real-world human-robot teaming in a field environment, autonomous robots often need to follow a human teammate to perform certain operations (e.g., following a skier while capturing videos and following a rescuer while carrying equipment as shown in Figure 3.1) in an unstructured and dynamic scenarios that evolve over time.

ROPA is a principled approach that formulates perceptual adaptation as a joint learning problem to simultaneously learn a base perception model to optimally fuse multisensory input data and a calibration term to adapt to environment changes. In order to fuse multisensory inputs, we implement sparsity-inducing norms that enforces the base perception model to learn sparse weights of multisensory input features and apply the weights for data fusion. To achieve perception calibration, we estimate the representativeness of the input feature modalities. Representativeness of a feature modality is referred to as its capability to represent the environment. When the environment changes, the representativeness of each feature modality also changes (e.g., depth can better represent dark environments). Accordingly, by automatically selecting feature modalities that are more representative in a specific environment, our approach provides a calibration of the perception model according to the environmental change. All the above components are mathematically integrated into a joint learning formulation under the unified theoretical framework of regularized optimization. In long-term human following, for each of new data instance, ROPA uses the joint base perception model and calibration term to classify humans under environment changes; classification results are applied by a decision making module to control the robot to navigate and follow the human. In order to evaluate ROPA and benchmark techniques for robot perceptual adaptation in human following applications, we collect a new large-scale dataset called *RObot Perceptual Adaptation* (ROPA). The dataset consists of multisensory perception input data collected from physical mobile robots in real-world field applications under short-term and long-term environment variations. Our experimental results over the PEAC dataset have validated that the proposed approach outperforms previous state-of-the-art methods for human following, obtains real-time performance, and is capable to address long-term and short-term environment changes.

The contribution of this paper[7] are:

[7] A preliminary non-archived version of the paper describing the dataset and initial experimental results was presented as a spotlight talk at the *ICRA Workshop on Robot Teammates Operating in Dynamic, Unstructured Environments* (RT-DUNE) [65].

- We propose a fresh human-inspired idea that addresses a new research problem of robot perceptual adaptation to short and long-term environment variations through calibrating robot perception to the current context in teammate following applications.
- We introduce ROPA that estimates the importance of heterogenous sensory data, integrates all information to build a perception model, and, more importantly, calibrates the perception model to adapt to short and long-term variations of the environment.
- We implement a new algorithm to solve the formulated optimization problem, which possesses a theoretical guarantee to converge to an optimal solution.
- As a practical contribution, we collect a new large-scale dataset from mobile robots, called *Perceptual Adaptation to Environment Changes* (PEAC), which includes three representative human-following scenarios of long-term autonomy in field applications to benchmark methods for robot adaptation to short and long-term environment changes.

The remainder of this paper is organized as follows. We review the related work on robot perception and adaptation in Section 3.3. The proposed adaptation approach is discussed in Section 5.4, and its optimization solver is described in 3.4.4 After describing the new dataset and our applications, we present and analyze the experimental results in Section 3.6. Finally, we conclude the paper in Section 3.8.

3.3 Related Work

In this section we provide a review of related research on long-term autonomy, robot adaptation, and human following. Long-term autonomy has received an increase in attention from the robotics community, because of increasing use of robots in environments presenting long-term dynamics (for example, robot following of humans in field environments throughout long periods of time can experience long-term changes). Robot adaptation is considered one viable solution to enable long-term autonomy, as adaptation enables the robot to cope with the changing environment.

3.3.1 Long-Term Autonomy

As robots are leaving factories and entering unstructured and dynamic environments, the ability of robots to reliably operate over long-periods of time under dynamically changing conditions needs to be addressed. Several learning-based approaches have been proposed to support robot long-term autonomy.

These approaches can be generally categorized into two paradigms: online learning and representation learning. (1) *Online learning* methods address long-term autonomy by continuously or iteratively updating model parameters during task execution [66–68]. The online learning paradigm is widely applied to a variety of applications with robots operating in dynamic or evolving environments [69], including health care, education, and assistive robotics in work [70] and home [71] environments. (2) *Representation learning* aims at learning from data to construct a representation of the robot's surrounding environment, which is robust or insensitive to environment variations. This paradigm is widely used in long-term place recognition (a.k.a. loop closure detection) to achieve Simultaneous Localization and Mapping (SLAM) in long-term settings ([72–75]). Most of the learning techniques for long-term place recognition focus on creating representations that encode the holistic layout of the environment based upon global feature extraction ([43]), deep learning [50, 76], multimodal feature integration [77], and spatio-temporal fusion [47]. Several recent methods use landmarks to create long-term environment representations [50, 57]. However, these methods are specifically implemented for place recognition or robot localization problems, and cannot be applied to appropriate perception of objects of interest under short and long-term context or environment changes. [78] evaluated YOLO's [79] performance during long-term changes on detecting an object of interest over long periods of time under various lighting conditions. It was observed that YOLO continuously struggles to detect the same object during sudden changes and during night time. Online learning methods lack the ability to address dramatic and fast changes. Although, they are effective in adapting to environments with slow and gradual changes, they fail under drastic changing conditions as they need time to converge.

3.3.2 Robot Adaptation

Although robot learning [80] has been addressed by many researchers, robot adaptation still remains to receive comparatively less attention. This is because many robot systems are designed to be used in very specific domains for a brief period of time [81, 82]. Early research focused on high-level behavior-based methods to address robot adaptation in general. For example, [83] developed various architectures to enable teams of heterogeneous robots to dynamically adapt their actions over time. Following a similar direction, case-based reasoning [58] methods were used in [16, 17] for robot behavioral adaptation in evolving environments. Another adaptation scheme was proposed in [21] to control complex robots, which selects a solution from a library of well-performing solutions, given specific tasks and conditions. In these early methods, robot adaptation is generally manually pre-determined, requiring significant domain expertise.

Adaptation based on human intent is studied in the domain of human-robot interaction and collaboration. One of the key components of this adaptation is to be able to recognize human intent and activities [84–86]. For example, a robot may need to recognize human intent and activities based upon visual feedback [87] or audio command [88]. Another popular learning-based adaptation paradigm is reinforcement learning, which is usually designed for robot behavior adaptation [22–24]. Recently, several methods [89–92] studied co-adaptation problems addressing how robots and humans on the same team can collaboratively adapt to each other and complete the joint task effectively. Almost all learning-based methods focus on behavior adaptation. The critical problem of robot perceptual adaptation has not been well understood and studied.

3.3.3 Human Following

A large portion of methods used in human following typically involve detecting humans [93], [79] and tracking humans [94], [95].

Researchers have tried to solve the problem of human following using different approaches. Most of these methods, in general, involve dividing the query image into several regions and using region proposals to predict the bounding boxes the human might be in [79], [96]. The core of most of these methods involve both feature extraction (to represent the regions from the bounding box in a different representation space) and then a classifier (to classify if the input feature representation of that region is of a human or not). Feature extraction methods can be further categorized into local features [97], [98], [99], [100] that describe the local information from different regions of interest, and global features [101], [38], [40] which generally describe the image as a whole. A global feature vector is generated based on the feature statistics. [101] is one of the most used global features for whole body human detection that captures the local shape and edge information of the whole image. [93] used optical flow field's internal difference to recognize moving humans. While local and global features can both be used to detect humans, global features have proved to give better results [102]. Recently, many methods use a combination of different sensors to achieve the capability of human detection in autonomous robots [103], [104]. It has also been proven that use of multi-sensory and multi-feature representation can significantly improve performance in long-term settings [54], [9]. Although these methods perform well in most scenarios, they cannot be considered in situations where the robot needs to adapt with the environment. Our approach can take in values from different sensors and calibrate their importance based on the environmental context, allowing robots the ability to adapt with the environment.

3.4 The ROPA Approach

To address multisensory robot perceptual adaptation in long-term autonomy, we propose the ROPA approach to learn and calibrate a perception model that can adapt to short-term and long-term environment changes.

Notation. Matrices are represented by boldface capital letters, and vectors by boldface lower-case letters. Given a matrix $\mathbf{U} = \{u_{ij}\} \in \Re^{n \times m}$, we represent the i-th row and j-th column as $\mathbf{u}^i$ and $\mathbf{u}_j$, respectively. The ℓ_2-norm of the vector $\mathbf{u}$ is defined as $\|\mathbf{u}\|_2 = \sqrt{\mathbf{u}^\top \mathbf{u}}$. The Frobenius norm of the matrix $\mathbf{U}$ is defined as $\|\mathbf{U}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n u_{ij}^2}$.

3.4.1 Multimodal Sensor Fusion

Given a collection of n data instances (i.e., data samples in a dataset), the extracted normalized feature vectors are represented as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \Re^{d \times n}$, where $\mathbf{x}_i \in \Re^d$ is the feature vector from i-th instance. We assume that the features are extracted from different robot sensors (e.g., color and depth), and various types of features are extracted, with each type named a *modality*. That is, a modality is the set of features that are computed using a feature extraction method from the input of a specific sensor. Then, each heterogeneous feature vector $\mathbf{x}_i \in \Re^d$ is assumed to consist of m-modalities of normalized features, such that $d = \sum_{j=1}^{m} d_j$. The label vector of classes associated with $\mathbf{X}$ is denoted by $\mathbf{Y} = [\mathbf{y}^1; \mathbf{y}^2; \ldots; \mathbf{y}^n] \in \Re^{n \times c}$, where c is the number of classes. Each element y_{ij} of the matrix $\mathbf{Y}$ indicates how likely the input feature vector $\mathbf{x}_i$ belongs to the j-th class and is manually labeled as the ground truth during the training phase.

Then, we formulate the multisensory recognition task as an optimization problem using the objective:

$$\min_{\mathbf{W}} \|\mathbf{Y} - (\mathbf{X}^\top \mathbf{W} + \mathbf{1}_n \mathbf{b}^\top)\|_F^2 \tag{3.1}$$

where $\mathbf{1}_n \in \Re^{n \times 1}$ is the constant vector of all 1's, $\mathbf{b} \in \Re^{c \times 1}$ is the bias vector that can be calculated by $\mathbf{b} = \mathbf{Y}^\top \mathbf{1}_n / n$. The solution to the optimization problem in Eq. (3.1) is a parameter matrix $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2 \ldots, \mathbf{w}_c] \in \Re^{d \times c}$, which consists of the weights $\mathbf{w}_i \in \Re^d$ of each element in the feature vector with respect to the i-th class.

Different types of features encode different attributes of the environment (e.g. shape, color, and edges). When fusing the features, some modalities are more informative than others depending on the robot operation environment, and it is desirable to estimate the importance of each modality. Inspired by sparse optimization [54], to identify discriminative modalities, we design a norm $\mathcal{R}_M(\mathbf{W})$ as a regularizer to Eq. (3.1), which enforces sparsity among the modalities and the grouping effect of the features within the same modality. The $\mathcal{R}_M$-norm can be expressed as $\mathcal{R}_M(\mathbf{W}) = \sum_{i=1}^{c} \sum_{j=1}^{m} \|\mathbf{w}_i^j\|_2$, which applies the ℓ_2-norm to the weights of feature elements within each modality and the ℓ_1-norm across different modalities. As $\mathcal{R}_M$ encodes the weight structure among modalities, we call it a modality norm.

As modern robots are usually equipped with a variety of sensors (e.g., color and depth), it is also desirable to estimate the importance of each sensor for multisensory robot perception. For example, a robot operating in dark can benefit more from depth sensors rather than color sensors. To meet this need, we introduce a sensory norm $\mathcal{R}_S$ to identify the discriminative sensors, which is defined as $\mathcal{R}_S(\mathbf{W}) = \sum_{i=1}^{c} \sum_{k=1}^{l} \|\mathbf{w}_i^k\|_2$. It applies the ℓ_2-norm to the weights of the features computed from the same sensor, and applies the ℓ_1-norm to the weights of features from different sensors.

By applying both modality and sensory norms, we formulate multisensory sensor fusion as a regularized optimization problem with the following objective function (where α is a trade-off hyperparameter):

$$\min_{\mathbf{W}} \|\mathbf{Y} - (\mathbf{X}^\top \mathbf{W} + \mathbf{1}_n \mathbf{b}^\top)\|_F^2 + \alpha\big(\mathcal{R}_M(\mathbf{W}) + \mathcal{R}_S(\mathbf{W})\big) \tag{3.2}$$

3.4.2 Perception Calibration

The key novelty of this paper is the introduction of the perception calibration capability, which is inspired by the psychology study on how human perception adapts in a given environmental context. Mathematically, we denote the environmental context as a matrix $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n] \in \Re^{s \times n}$, where $\mathbf{e}_i = [e_i^1, e_i^2, \dots, e_i^s]^\top \in \Re^s$ is a low-dimensional vector consisting of s environmental context variables (e.g., lighting, fog intensity, and ground traction) obtained along with the i-th data instance.

To achieve perception calibration, we estimate the representativeness of each feature to represent the environmental context. Representativeness of a feature is referred to as its capability of representing the environment. When the environmental context changes (e.g., lighting variations), we can estimate the feature representativeness change to still represent the environment under such context changes. Therefore, estimating the feature representativeness change encodes our insight of calibrating the perception model (i.e., dynamically adjusting the weights of the features) according to the context. To achieve our insight, two procedures need to be performed: (1) computing the representativeness of the features to represent the environment in each instance, and

(2) associating the computed feature representativeness with the context variables.

We denote the feature representativeness to represent the environment in the instances as $\mathbf{G} = [\mathbf{g}^1, \mathbf{g}^2, \ldots, \mathbf{g}^n]^\top \in \Re^{n \times d}$. Then, each $\mathbf{g}^i \in \Re^d$ denotes the representativeness of the feature vector $\mathbf{x}_i$ to represent the environment, which can be estimated by $\mathbf{g}^i \mathbf{x}_i = 1$. By computing the Moore-Penrose inverse (a.k.a. psuedo-inverse), we can obtain $\mathbf{g}^i = \mathbf{x}_i^\top \left(\mathbf{x}_i \mathbf{x}_i^\top\right)^{-1}$.

We associate the representativeness matrix $\mathbf{G}$ (of the features to represent the environment) with the context variables through designing a novel loss function $\mathcal{L}(\mathbf{G}, \mathbf{E}; \mathbf{V})$, with the objective of learning a projection (parameterized by $\mathbf{V}$) from the context variables $\mathbf{E}$ to encode $\mathbf{G}$. Specifically, the loss function $\mathcal{L}$ is defined as:

$$\mathcal{L}(\mathbf{G}, \mathbf{E}; \mathbf{V}) = \|\mathbf{G} - \mathbf{E}^\top \mathbf{V}\|_F^2 \tag{3.3}$$

where the parameter matrix $\mathbf{V} \in \Re^{s \times d}$ includes the weights of the environment context variables with respect to the features.

That is, $\mathbf{V}$ captures the underlying information of how much each of the elements in the feature vectors should change, given the change in the environmental context.

Similar to the motivation of estimating the importance of features and sensors, it is desirable to learn the importance of the environment context variables to calibrate the perception model. Therefore, we develop the new context norm $\mathcal{R}$ over the parameter matrix $\mathbf{V}$, defined as $\mathcal{R}_C(\mathbf{V}) = \sum_{p=1}^{s} \|\mathbf{v}^p\|_2$, which enforces the sparsity between the environment context variables with respect to all features.

Therefore, to adapt to short-term and long-term environment changes, we integrate the proposed perceptual calibration capability with multisensory fusion. We formulate multisensory perceptual adaptation as a joint learning problem under the unified regularized optimization framework, with the final objective function:

$$\min_{\mathbf{W}, \mathbf{V}} \quad \|\mathbf{Y} - (\mathbf{X}^\top \mathbf{W} + \mathbf{1}_n \mathbf{b}^\top)\|_F^2 + \|\mathbf{G} - \mathbf{E}^\top \mathbf{V}\|_F^2 + \alpha\big(\mathcal{R}_M(\mathbf{W} + \mathcal{R}_S(\mathbf{W})\big) + \beta \mathcal{R}_C(\mathbf{V}) \tag{3.4}$$

where β is a trade-off hyperparameter.

3.4.3 Multisensory Robot Perceptual Adaptation

After solving the formulated regularized optimization problem in Eq. (3.4) during training (using the solver in Algorithm 3), we obtain the optimal $\mathbf{W}^* = [\mathbf{w}_1^*, \mathbf{w}_2^*, \ldots, \mathbf{w}_c^*]$ and $\mathbf{V}^*$. Then, during online execution, given a newly acquired data instance $\mathbf{x}$ and its environmental context variable $\mathbf{e}$, adaptive robot perception to determine the class label $y(\mathbf{x}, \mathbf{e})$ can be performed by:

$$y(\mathbf{x}, \mathbf{e}) = \max_i (\mathbf{x}^\top)\big(\operatorname{diag}(\mathbf{e}^\top \mathbf{V}^*)\mathbf{w}_i^* + \mathbf{w}_i^*\big) + b_i \tag{3.5}$$

where $\operatorname{diag}(\cdot)$ denotes a function to covert a vector into a diagonal matrix. Given any vector $\mathbf{a} \in \Re^z$, $\operatorname{diag}(\mathbf{a}) = \mathbf{a} \times \mathbf{I}_{z \times z}$, where $\mathbf{I}$ is an identity matrix. The output of Eq. (3.5) provides the decision of whether a human is present or not. Given a query feature vector $\mathbf{x}$, and its associated environmental context $\mathbf{e}$, the term $\operatorname{diag}(\mathbf{e}^\top \mathbf{V}^*)\mathbf{w}_i^*$ provides an estimation of the calibration needed to adjust each feature weight given $\mathbf{e}$ for the query $\mathbf{x}$. This calibration is then added to the feature weights $\mathbf{w}$ to determine the class label.

One of the advantages of our approach is that classification is integrated with feature learning and model calibration under the unified regularized optimization framework, thus eliminating the requirement of using additional classifiers. In addition, our formulation is based on convex linear models, which makes model parameter estimation and online inference highly efficient. Thus, ROPA is able to achieve high-speed onboard processing, which can significantly benefit real-time robotics applications.

3.4.4 Optimization Algorithm

Although our formulation is convex, the objective function in Eq. (3.4) is difficult to solve in general because of the three non-smooth regularizers. Another contribution of this paper is that we implement an iterative algorithm, that follows the iterative re-weighted method [55], to solve the optimization problem,which is presented in Algorithm 3.

Algorithm 2: An iterative algorithm to solve the formulated optimization problem in Eq. (3.4)

```
Input  : Feature matrix $\mathbf{X} \in \Re^{d \times n}$, label matrix $\mathbf{Y} \in \Re^{n \times c}$, context variables $\mathbf{E} \in \Re^{s \times n}$
Calculate the representativeness matrix $\mathbf{G}$ from $\mathbf{X}$.
Let $t = 1$. Initialize $\mathbf{W}(t)$ and $\mathbf{V}(t)$ by solving $\min_{\mathbf{W}} \|\mathbf{Y} - (\mathbf{X}^\top \mathbf{W} + \mathbf{1}_n \mathbf{b}^\top)\|_F^2$ and
  $\min_{\mathbf{V}} \|\mathbf{G} - \mathbf{E}^\top \mathbf{V}\|_F^2$.
while not converge do
    Calculate the block diagonal matrix $\mathbf{D}^i(t+1)(1 \le i \le c)$, where the $j$-th block is $\frac{1}{2\|\mathbf{w}_i^j(t)\|_2}\mathbf{I}_j$.
    Calculate the block diagonal matrix $\widehat{\mathbf{D}^i}(t+1)(1 \le i \le c)$, where the $k$-th block is $\frac{1}{2\|\mathbf{w}_i^k(t)\|_2}\mathbf{I}_k$.
    Calculate the block diagonal matrix $\widetilde{\mathbf{D}}^j(t+1)(1 \le j \le d)$, where the $p$-th block is $\frac{1}{2\|\mathbf{v}_j^p(t)\|_2}\mathbf{I}_p$.
    For each $\mathbf{w}_i(1 \le i \le c)$, $\mathbf{w}_i(t+1) = (\mathbf{X}\mathbf{X}^\top + \alpha\mathbf{D}^i(t+1) + \alpha\widehat{\mathbf{D}^i}(t+1))^{-1}\mathbf{X}(\mathbf{y}_i - \mathbf{b}_i)$.
    For each $\mathbf{v}_j(1 \le j \le d)$, $\mathbf{v}_d = (\mathbf{E}\mathbf{E}^\top + \beta\widetilde{\mathbf{D}}^j(t+1))^{-1}\mathbf{E}\mathbf{g}_j$
    $t = t + 1$.
Output: $\mathbf{W} = \mathbf{W}(t) \in \Re^{p \times c}$ ;$\mathbf{V} = \mathbf{V}(t) \in \Re^{s \times d}$
```

To learn the optimal $\mathbf{W}$, we compute the derivative of the objective function in Eq. (3.4) with respect to $\mathbf{w}_i(1 \le i \le c)$ and set it to a zero vector, as follows:

$$\mathbf{X}\mathbf{X}^\top \mathbf{w}_i - \mathbf{X}\left(\mathbf{y}_i - \mathbf{b}_i\right) + \alpha\mathbf{D}^i\mathbf{w}_i + \alpha\widehat{\mathbf{D}}^i\mathbf{w}_i = \mathbf{0} \tag{3.6}$$

where $\mathbf{D}^i(1 \le i \le c)$ is a block diagonal matrix with j-th diagonal block computed by $\frac{1}{2\|\mathbf{w}_i^j\|_2}\mathbf{I}_j$; $\mathbf{w}_i^j$ is the j-th segment of $\mathbf{w}_i$ consisting of the weights from the j-th feature modality; $\widehat{\mathbf{D}}^i$ is a block diagonal matrix with the k-th diagonal block computed by $\frac{1}{2\|\mathbf{w}_i^k\|_2}\mathbf{I}_k$; and $\mathbf{w}_i^k$ is the k-th segment of $\mathbf{w}_i$ consisting of the weights of features from k-th modality. After solving Eq. (3.6), the vector $\mathbf{w}_i$ can be computed by:

$$\mathbf{w}_i = \left(\mathbf{X}\mathbf{X}^\top + \alpha\mathbf{D}^i + \alpha\widehat{\mathbf{D}}^i\right)^{-1}\mathbf{X}(\mathbf{y}_i - \mathbf{b}_i) \tag{3.7}$$

To compute the optimal value for $\mathbf{V}$, compute the derivative of the objective function in Eq. (3.4) with respect to the columns $\mathbf{v}_j, (1 \le j \le d)$ of $\mathbf{V}$ and set the resulting expression to zero, as follows:

$$\mathbf{E}\mathbf{E}^\top \mathbf{v}_j - \mathbf{E}\mathbf{g}_j + \beta\widetilde{\mathbf{D}}^j\mathbf{v}_j = \mathbf{0} \tag{3.8}$$

where $\widetilde{\mathbf{D}}^i$ is a block diagonal matrix with j-th block computed by $\frac{1}{2\|\mathbf{v}_j^p\|_2}\mathbf{I}_p$; $\mathbf{v}_j^p$ is the p-th segment of $\mathbf{v}_j$ that specifies the weights of the p-th environmental context variable with respect to the j-th feature modality. Solving the above equation, we can obtain:

$$\mathbf{v}_j = \left(\mathbf{E}\mathbf{E}^\top + \beta\widetilde{\mathbf{D}}^j\right)^{-1}\mathbf{E}\mathbf{g}_j \tag{3.9}$$

Because $\mathbf{D}^i$ and $\widehat{\mathbf{D}}^i$ depend on $\mathbf{W}$, and because $\widetilde{\mathbf{D}}^j$ depends on $\mathbf{V}$, $\mathbf{D}^i$, $\widehat{\mathbf{D}}^i$, and $\widetilde{\mathbf{D}}^j$ are also unknown variables. Therefore, we design and implement an iterative algorithm to solve this optimization problem, which is presented in Algorithm 2. The proposed optimization solver holds a theoretical convergence guarantee to an optimum, as described by Theorem 2.

Theorem 2: Algorithm 2 converges to an optimal solution to the optimization problem in Eq. (3.4)

Proof. In the following, we prove that Algorithm 2 decreases the value of the objective function with each iteration and converges to an optimal value. But first, we present a lemma:

Lemma 2: For any two given vectors $\mathbf{a}$ and $\mathbf{b}$, the following inequality relation holds: $\|\mathbf{b}\|_2 - \frac{\|\mathbf{b}\|_2^2}{2\|\mathbf{a}\|_2} \leq \|\mathbf{a}\|_2 - \frac{\|\mathbf{a}\|_2^2}{2\|\mathbf{a}\|_2}$

Proof. We have:

$$-(\|\mathbf{b}\|_2 - \|\mathbf{a}\|_2)^2 \leq 0$$

$$-\|\mathbf{b}\|_2^2 - \|\mathbf{a}\|_2^2 + 2\|\mathbf{b}\|_2\|\mathbf{a}\|_2 \leq 0$$

$$2\|\mathbf{b}\|_2\|\mathbf{a}\|_2 - \|\mathbf{b}\|_2^2 \leq \|\mathbf{a}\|_2^2$$

$$\|\mathbf{b}\|_2 - \frac{\|\mathbf{b}\|_2^2}{2\|\mathbf{a}\|_2} \leq \|\mathbf{a}\|_2 - \frac{\|\mathbf{a}\|_2^2}{2\|\mathbf{a}\|_2}$$

□

From Algorithm 2, we know that:

$$\begin{aligned}\mathbf{W}(t+1) = \min_{\mathbf{W}} \|\mathbf{Y} - (\mathbf{X}^\top\mathbf{W} + \mathbf{1}_n\mathbf{b}^\top)\|_F^2 + \alpha\sum_{i=1}^{c}\mathbf{w}_i^\top(t+1)\mathbf{D}^i(t+1)\mathbf{w}_i(t+1) \\ + \quad \alpha\sum_{i=1}^{c}\mathbf{w}_i^\top(t+1)\widehat{\mathbf{D}}^i(t+1)\mathbf{w}_i(t+1)\end{aligned} \tag{3.10}$$

and,

$$\mathbf{V}(t+1) = \min_{\mathbf{V}} \|\mathbf{G} - \mathbf{E}^{\top}\mathbf{V}\|_F^2 + \beta \sum_{j=1}^{d} \mathbf{v}_j^{\top}(t+1)\widetilde{\mathbf{D}}^j(t+1)\mathbf{v}_j(t+1) \tag{3.11}$$

Then it can be derived that:

$$\begin{aligned} &\mathcal{F}(t+1) + \alpha \sum_{i=1}^{c} \mathbf{w}_i^{\top}(t+1)\mathbf{D}^i(t+1)\mathbf{w}_i(t+1) + \alpha \sum_{i=1}^{c} \mathbf{w}_i^{\top}(t+1)\widehat{\mathbf{D}}^i(t)\mathbf{w}_i(t+1) \\ &\qquad \le \mathcal{F}(t) + \alpha \sum_{i=1}^{c} \mathbf{w}_i^{\top}(t)\mathbf{D}^i(t)\mathbf{w}_i(t) + \alpha \sum_{i=1}^{c} \mathbf{w}_i^{\top}(t)\widehat{\mathbf{D}}^i(t)\mathbf{w}_i(t) \end{aligned} \tag{3.12}$$

and,

$$\mathcal{J}(t+1) + \beta \sum_{j=1}^{d} \mathbf{v}_j^{\top}(t+1)\widetilde{\mathbf{D}}^j(t+1)\mathbf{v}_j(t+1) \le \mathcal{J}(t) + \beta \sum_{j=1}^{d} \mathbf{v}_j^{\top}(t)\widetilde{\mathbf{D}}^j(t)\mathbf{v}_j(t) \tag{3.13}$$

Substituting the values of $\mathbf{D}^i$, $\widehat{\mathbf{D}}^i$ and $\widetilde{\mathbf{D}}^j$, we get:

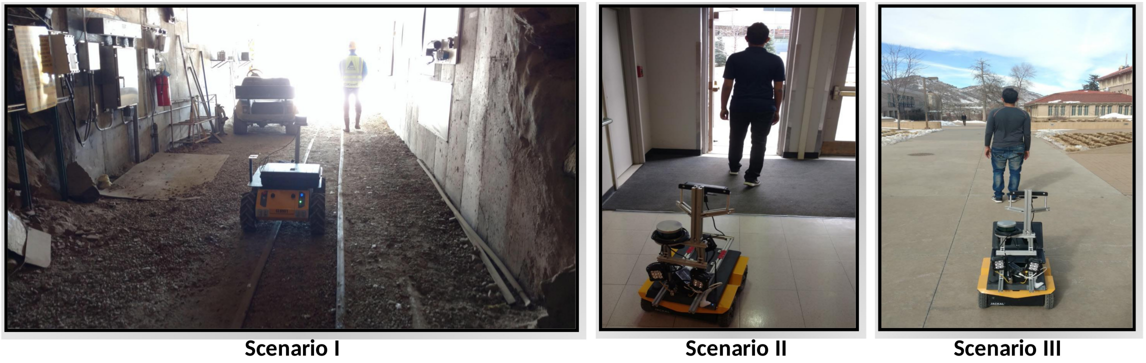
$$\begin{aligned} &\mathcal{F}(t+1) + \alpha \sum_{i=1}^{c}\sum_{j=1}^{m} \frac{\|\mathbf{w}_i^j(t+1)\|_2^2}{2\|\mathbf{w}_i^j(t)\|_2} + \alpha \sum_{i=1}^{c}\sum_{k=1}^{l} \frac{\|\mathbf{w}_i^k(t+1)\|_2^2}{2\|\mathbf{w}_i^k(t)\|_2} \\ &\qquad \le \mathcal{F}(t) + \alpha \sum_{i=1}^{c}\sum_{j=1}^{m} \frac{\|\mathbf{w}_i^j(t)\|_2^2}{2\|\mathbf{w}_i^j(t)\|_2} + \alpha \sum_{i=1}^{c}\sum_{k=1}^{l} \frac{\|\mathbf{w}_i^k(t)\|_2^2}{2\|\mathbf{w}_i^k(t)\|_2} \end{aligned} \tag{3.14}$$

and,

$$\mathcal{J}(t+1) + \beta \sum_{p=1}^{s}\sum_{j=1}^{d} \frac{\|\mathbf{v}_j^p(t+1)\|_2^2}{2\|\mathbf{v}_j^p(t)\|_2} \quad \le \quad \mathcal{J}(t) + \beta \sum_{p=1}^{s}\sum_{j=1}^{d} \frac{\|\mathbf{v}_j^p(t)\|_2^2}{2\|\mathbf{v}_j^p(t)\|_2} \tag{3.15}$$

From Lemma 6, we can derive the following equations:

$$\sum_{i=1}^{c}\sum_{j=1}^{m} \|\mathbf{w}_i^j\|_2 - \sum_{i=1}^{c}\sum_{j=1}^{m} \frac{\|\mathbf{w}_i^j(t+1)\|_2^2}{2\|\mathbf{w}_i^j(t)\|_2} \le \sum_{i=1}^{c}\sum_{j=1}^{m} \|\mathbf{w}_i^j\|_2 - \sum_{i=1}^{c}\sum_{j=1}^{m} \frac{\|\mathbf{w}_i^j(t)\|_2^2}{2\|\mathbf{w}_i^j(t)\|_2} \tag{3.16}$$



Scenario I Scenario II Scenario III

Figure 3.2 Illustration of the scenarios in which the PEAC dataset was collected for the task of human teammate following.

$$\sum_{i=1}^{c}\sum_{k=1}^{l}\|\mathbf{w}_i^k\|_2 - \sum_{i=1}^{c}\sum_{k=1}^{l}\frac{\|\mathbf{w}_i^k(t+1)\|_2^2}{2\|\mathbf{w}_i^k(t)\|_2} \leq \sum_{i=1}^{c}\sum_{k=1}^{l}\|\mathbf{w}_i^k\|_2 - \sum_{i=1}^{c}\sum_{k=1}^{l}\frac{\|\mathbf{w}_i^k(t)\|_2^2}{2\|\mathbf{w}_i^k(t)\|_2} \tag{3.17}$$

$$\sum_{p=1}^{s}\sum_{j=1}^{d}\|\mathbf{v}_j^p\|_2 - \sum_{p=1}^{s}\sum_{j=1}^{d}\frac{\|\mathbf{v}_j^p(t+1)\|_2^2}{2\|\mathbf{v}_j^p(t)\|_2} \leq \sum_{p=1}^{s}\sum_{j=1}^{d}\|\mathbf{v}_j^p\|_2 - \sum_{p=1}^{s}\sum_{j=1}^{d}\frac{\|\mathbf{v}_j^p(t)\|_2^2}{2\|\mathbf{v}_j^p(t)\|_2} \tag{3.18}$$

Adding Equations (3.14) - (3.18) on both the side we get that

$$\begin{aligned}\mathcal{F}(t+1) + \mathcal{J}(t+1) + \alpha\sum_{i=1}^{c}\sum_{j=1}^{m}\|\mathbf{w}_i^j(t+1)\|_2 + \alpha\sum_{i=1}^{c}\sum_{k=1}^{l}\|\mathbf{w}_i^k(t+1)\|_2 \\ +\sum_{p=1}^{s}\sum_{j=1}^{d}\beta\|\mathbf{v}_j^p(t+1)\|_2 \leq \mathcal{F}(t) + \mathcal{J}(t) + \alpha\sum_{i=1}^{c}\sum_{j=1}^{m}\|\mathbf{w}_i^j(t)\|_2 \\ +\alpha\sum_{i=1}^{c}\sum_{k=1}^{l}\|\mathbf{w}_i^k(t)\|_2 + \sum_{p=1}^{s}\sum_{j=1}^{d}\beta\|\mathbf{v}_j^p(t)\|_2\end{aligned} \tag{3.19}$$

Eq. (3.19) proves that the value of the objective function decreases in each iteration. Because the formulated objective function is convex, Algorithm 2 converges to an optimal solution. □

Complexity: Since the optimization problem in Eq. (3.4) is convex, Algorithm 2 converges to a local optimal solution fast. In each iteration of our algorithm, computing Steps 4-6 is trivial. We compute Steps 7 and 8 by solving a system of linear equations with a quadratic complexity.

3.5 Dataset and Implementation

3.5.1 The PEAC Dataset

A practical contribution of this research is the collection of a new dataset of Perceptual Adaptation to Environment Changes (PEAC). Although long-term autonomy has been recently attracting an increasing attention in robotics, before this work, no dataset is publicly available for benchmarking robot perceptual adaptation, which consists of multisensory perception data collected from physical robots in real-world field applications under short-term and long-term lighting changes. Motivated by this desire, we collected the PEAC dataset.We utilized the Clearpath Husky and Jackal mobile robots (shown in Figure 3.2) to follow and detect an individual human subject. The robots are equipped with the Asus Xtion PRO structured-light camera without any automatic gains to collect color-depth data, and the Adafruit TSL2561 digital luminosity sensor to collect luminosity data. The color and depth images have a resolution of 640×480. The luminosity readings is normalized between 0 (no lighting intensity) to 1 (maximum lighting intensity). Both structure-light camera and luminosity sensor run at 30 frames per second.

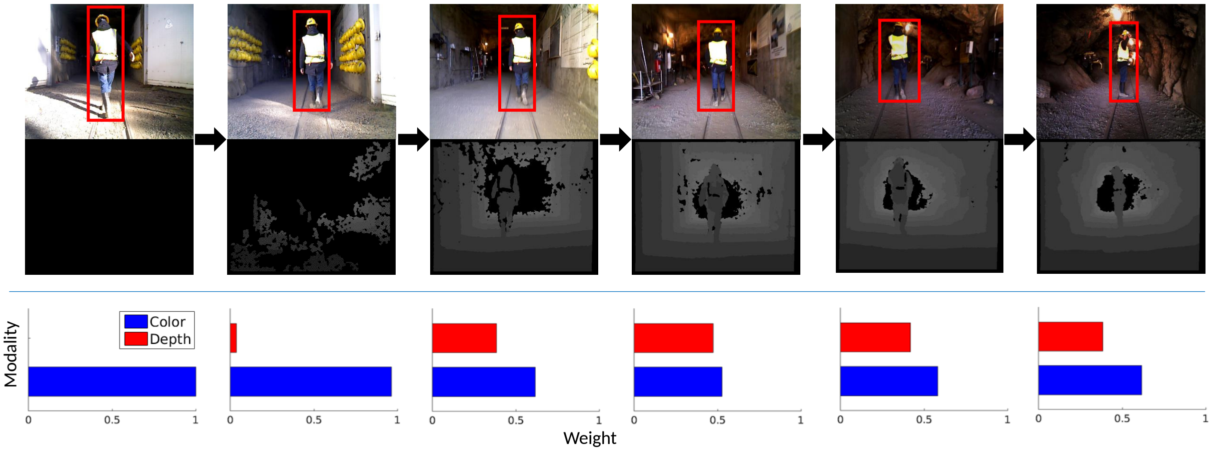


Figure 3.3 Qualitative and quantitative results on Scenario I (Mine-Entering-Exiting). The top row depicts an example of the qualitative results from the robot's viewpoint when it follows a human subject to navigate into a dark mine drift from a bright open area. The bottom row illustrates the importance of color-depth sensor modalities learned and adapted by ROPA.

During data collection, we assume the robot is performing a human following task, which is a desired capability in robotics applications (e.g., to carry rescue gear for humans in a long-term search and rescue operation). The robots are manually controlled by a separate human operator to follow a single human teammate walking in front of the robot. The dataset is collected in three different scenarios shown in Figure 3.2:

- Scenario I (Mine-Entering-Exiting): A mobile robot follows a human subject entering and exiting two different mine drifts (i.e., horizontal openings made in a mine). One drift is dark (Figure 3.2), and the other has light bulbs installed in the drift (Figure 3.3). When the robot travels from the inside to the outside of the mine drift (or vise versa), the environment exhibits significant lighting changes. This testing scenario represents possible situations when robot performs underground search and rescue, for example, in mine, cave, and subway environments.

- Scenario II (Indoor-Outdoor-Traveling): A robot follows an individual human to travel inside and outside of a building. This scenario may happen in robot-assisted daily living and/or urban search and rescue applications. Similar to Scenario I, this scenario contains the challenge of fast illumination changes when the robot enters and leaves the building, as shown in Figure 3.4.

- Scenario III (All-Day-Following): A mobile robot follows a human subject in an outdoor environment in different times of then day from dawn to dusk, and across different months. The environment changes dramatically, especially lighting levels from noon to late evening (Figure 3.5). In the daytime under strong sunshine, structured-light depth sensors fail. In evenings with poor lighting conditions, color cameras do not work well. This Scenario III includes significant challenges caused by long-term environment changes across a day and seasons.

In each scenario, we collect 20 instances of human following, each consisting of 700-1000 color-depth images. In addition, we collect the light level data (in lux) using digital luminosity sensors installed on the robots to document lighting variations in the environments. The ground truth of human detection is manually labeled by bounding human subjects in the color-depth scene with a box. The PEAC dataset is publicly available at http://hcr.mines.edu/code/PEAC.html.

3.5.2 Implementation

Three real-time feature extraction methods were applied to compute heterogeneous features from the proposed regions of the color and depth images readings in our experiments. (1) Histogram of Oriented Gradients (HOG) features ([101]) are used to represent shapes by counting occurrences of gradient orientation; (2) GIST features ([38]) are build from the response of steerable filters at different orientations and scales; and Local difference Binary Patterns (LBP) features ([40]) compute binary strings from simple intensity and gradient differences of image grid cells. These three features are selected because they are widely used in previous work on long-term place recognition. Each modality of features from color or depth data is normalized to ensure that features in the modalities have the same value range. To ensure real-time performance, we don't use deep features. But in principle, any features that can produce a vector-based representation can be applied as an input modality to work with ROPA. The luminosity data is adopted to represent the context, i.e., the lighting variation. Object proposals are provided by a proposal generation approach applied on both color and depth data ([96]) to obtain regions of interest that potentially contain humans. The current implementation of ROPA is programmed using a mixture of unoptimized Matlab and C++ on an onboard Linux computer within the Jackal and Husky robots, which has an i5 2.5 GHz CPU and 8G memory, with no GPU.

Given the detection results of humans from ROPA that addresses short-term and long-term environmental changes, human following is performed by a decision making module for robot navigation control. The module used in this work is implemented using an apprenticeship learning approach in the general framework of Markov decision process (MDP), which is detailed in our previous paper ([36]), which leverages human demonstrations to learn a navigation policy (e.g., move forward or backward, turn left or right, stop, speed up, slow down, etc.) based on human detection results.

3.6 Experiments

In this section, we present and analyze experimental results. To evaluate the performance of ROPA for multisensory robot perceptual adaptation, we performed extensive experiments on the PEAC dataset in the human following task.

To estimate ROPA's parameters, we employ around 4000 color-depth frames from Scenario I (Mine-Entering-Exiting) *only* for training. Features extracted from color-depth images using different techniques are concatenated into a final vector as the input to ROPA. The luminosity sensor reading is used to encode the context, i.e., environment lighting changes. Throughout our experiments, we set the values of the trade-off hyperparameters α and β to 0.1.

In the testing phase, we apply the trained model to the 5 instances from Scenario I (Mine-Entering-Exiting), and the same model to instances from Scenario II (Indoor-Outdoor-Traveling) and Scenario III (All-Day-Following). That is, no instances from Scenarios II and III are used for training. So we can evaluate how the ROPA model can be scaled to previously unseen situations.

To show the advantage of ROPA, we first compare ROPA with baseline techniques. We implement baselines based upon single sensor and single feature type (i.e., HOG, LBP, or GIST from color or depth sensor). The decision model in Eq. (3.5) without the calibration term is applied along with these features to recognize humans. In addition, we compare ROPA with previous state-of-the-art methods, including (1) Multimodal convolutional neural network (mCNN) ([105]), (2) Shared representative appearance learning (SRAL) ([54]), and (3) You only look once ([79]). The used techniques except YOLO are trained on the same set of data from the PEAC dataset. We use the YOLO model pretrained by the YOLO's authors under various illumination conditions ([106]).

3.6.1 Scenario I (Mine-Entering-Exiting)

The qualitative results obtained by ROPA are illustrated in Figure 3.3, which shows that ROPA allows the robot to accurately recognize the human when navigating from a bright outdoor open area into a dark mine drift under dramatic illumination changes.

Table 3.1 Comparison of average accuracy over all robot operation scenarios in the PEAC dataset

Methods	Scenario-I	Scenario-II	Scenario-III
HOG	89.22%	79.00%	27.74%
LBP	81.40%	71.95%	44.83%
GIST	79.53%	71.12%	55.10%
HOG-D	70.11%	65.16%	46.84%
LBP-D	61.07%	70.10%	47.45%
GIST-D	67.57%	48.98%	51.48%
YOLO	88.13%	77.65%	47.08%
SRAL	92.30%	85.07%	51.13%
mCNN	90.65%	87.24%	49.45%
ROPA	**96.91%**	**89.72%**	**79.17%**

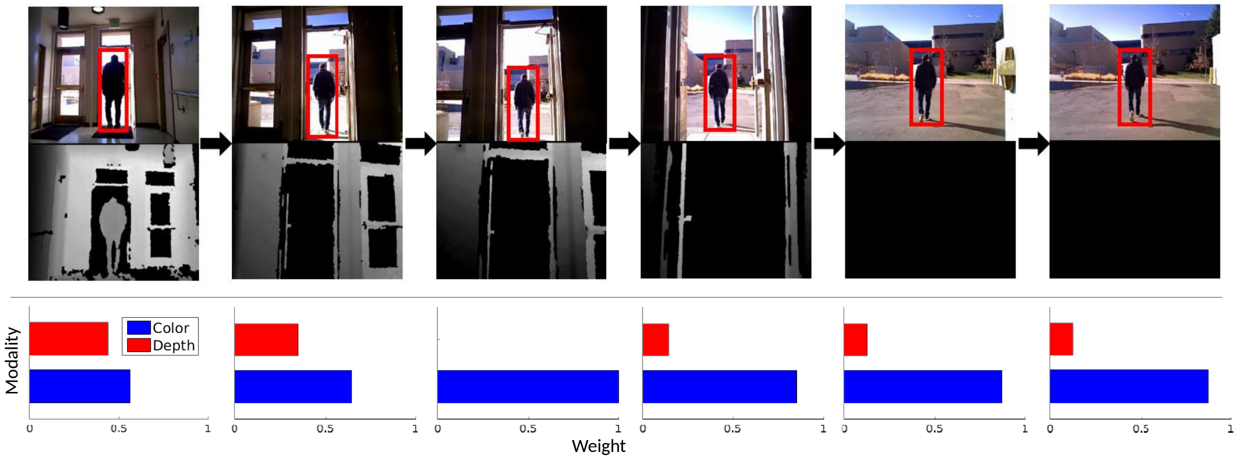


Figure 3.4 Qualitative and quantitative results on Scenario II (Indoor-Outdoor-Traveling). The top row illustrates an example of the recognition results from the robot's viewpoint when it follows the human to navigate from the inside to the outside of a building. The bottom row shows the importance of color-depth sensor modalities learned and adapted by our ROPA algorithm.

In addition, we compute numerical results to quantitatively evaluate ROPA. To evaluate how well ROPA can identify humans under lighting changes, accuracy is employed as an evaluation metric. Our ROPA obtains an average accuracy of 96.91% in Scenario I. We also compared ROPA with baseline and previous methods in Scenario I. The comparison results are presented in Table Table 3.1. It is observed that techniques based on color features (i.e., HOG, LBP, GIST, and YOLO) generally perform better than methods using depth features (i.e., HOG-D, LBP-D, and GIST-D).

Also, through integrating multisensory multimodal features, sensor fusion approaches (i.e., SRAL and mCNN) can outperform techniques using single types of features. ROPA significantly improves performance and obtains the best accuracy, due to its capability to immediately calibrate multisensory perception and adapt to environment changes.

Besides accuracy, we also evaluate how ROPA can adapt to environment changes by analyzing the importance of different sensor modalities, i.e., color and depth in this experiment. The quantitative results are graphically presented in Figure 3.3. It is observed that, when the robot stays outside of the mine under direct sunshine, it completely relies on color cues to recognize the subject because the structured-light depth sensor on the robots fail under direct sunshine, and cannot provide depth information. ROPA can automatically learn this fact from data without the requirement of hard coding. When the robot follows the human subject into the mine drift with reduced lighting, we observe that ROPA starts using the depth information to combine with color cues to recognize humans, which demonstrates ROPA's on-the-fly multisensory perception calibration capability to adapt to environment changes.

3.6.2 Scenario II (Indoor-Outdoor-Traveling)

We further evaluate ROPA's performance in Scenario II, in which a robot identifies and follows a human subject to travel between the inside and outside of a building. In this scenario, the robot needs to address the challenge of adapting to lighting differences, when it travels from the inside of a building to the outside (or vise versa). In the experiment, the perception model learned in Scenario I is directly applied to Scenario II without re-training or additional training, in order to show the ROPA's capability of adapting from one scenario to a new, previously unexperienced scenario.

The qualitative results obtained by ROPA over Scenario II are shown in Figure 3.4, which demonstrates that ROPA can recognize humans while following the subjects traveling from the inside of a building with low lighting levels into a bright outdoor environment. Quantitatively, ROPA obtains an average accuracy of 89.72% in identifying human subjects. In addition, we compare our ROPA approach with the baseline techniques, and present the results in Table 1 of the main paper. Consistent with the results observed in Scenario I, methods based on color cues perform better than methods that employ depth features only. In indoor environments with a reduced lighting level (but not completely dark), color cues still contribute to human recognition. When multisensory multimodal features are fused together, SRAL and mCNN obtain an improved accuracy over the baseline techniques based on a single type of features only. Because of the capability of calibrating multisensory perception, ROPA adapts to environment changes and outperforms the baseline and previous approaches. In addition, Figure 3.4 graphically illustrates the importance of the color and depth sensor modalities in the experiment. It can be observed that as the robot starts to navigate toward the outside of the building, color cues start to become more critical to recognize the human subject. The changes of the sensor weights match the variations of the surrounding environment, which demonstrates the multisensory perceptual adaptation capability of the robot enabled by ROPA.

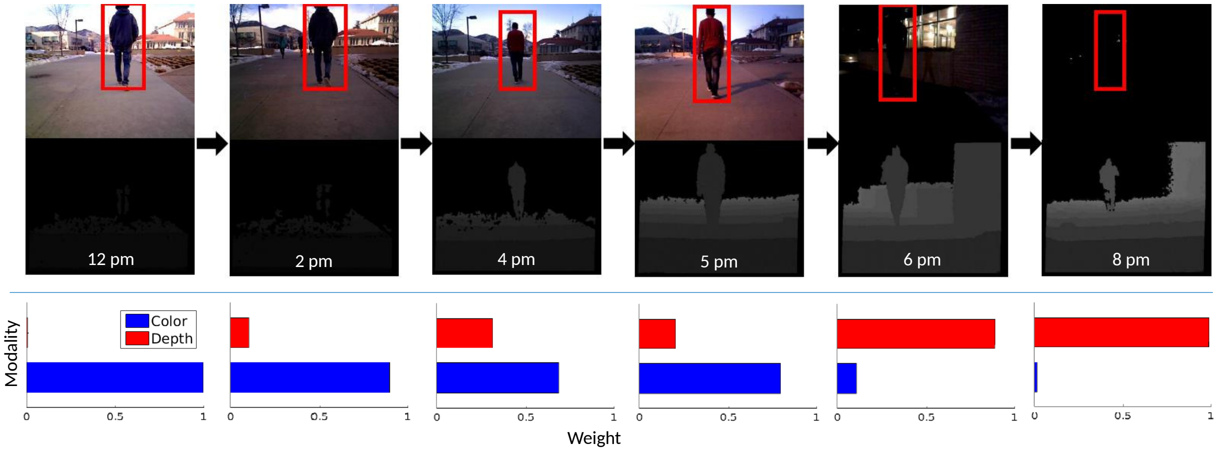


Figure 3.5 Qualitative and quantitative results on Scenario III (All-Day-Following). The top row illustrates an example of the qualitative results from the robot's viewpoint when it follows a subject to navigate in a campus environment from noon till late evening with dramatic long-term lighting changes. The bottom row shows the importance of color-depth sensor modalities learned and adapted by ROPA.

3.6.3 Scenario III (All-Day-Following)

To evaluate our approach's multisensory perception adaptation capability to *long-term* environment variations, we evaluate ROPA in Scenario III, in which a mobile robot follows a human to navigate in outdoor campus environments at different time of the day (e.g., from morning, noon, afternoon, till evening). In this experiment, the environment shows significant variations in different hours of the day during long-term robot operations. The perception model learned in Scenario I is directly used in this new long-term scenario without re-training, in order to demonstrate the ROPA's ability to calibrate perception automatically.

The qualitative results obtained by ROPA are illustrated in Figure 3.5, which demonstrates ROPA's capability to recognize humans under long-term lighting changes during the day. Quantitatively, ROPA obtains an average accuracy of 79.17%. With no re-training or incremental training, ROPA can still recognize humans well. But in general, the accuracy is lower than the scenarios with short-term environment changes. The comparison of our ROPA approach with baseline and previous state-of-the-art methods is listed in Table Table 3.1. An interesting observation is that the SRAL and mCNN approaches based on sensor fusion perform worse than some baseline techniques using a single type of features. This phenomenon results from the fact that for most of the time in this scenario, only color or only depth information is available (Figure 3.5). For example, under the sun, the depth sensor often fails, while during or after sunset, the color sensor does not operate well. In this scenario, sensor fusion paradigms may not be able to work well.

Adaptation of the importance weights of color and depth modalities is illustrated in Figure 3.5. The results validate that ROPA continuously calibrates multisensory robot perception to the surrounding environment with long-term changes. When the sunshine is strong (e.g., at the noon time when the depth sensor does not work), ROPA adapts to utilize color cues only for perception. On the other hand, when the environment is very dark (e.g., at 8PM), the color camera does not work, and ROPA can automatically calibrate perception to depend on depth cues for recognition and following. Another interesting observation from the experimental results is that, the weight of the color sensor at 5PM is greater than the weight at 4PM, although the environment has a decreased *natural* lighting. This occurs because the street lights were turned on right before 5PM, which provides additional artificial lighting to the scene, making the overall lighting on the pedestrian path better than that at 4PM.

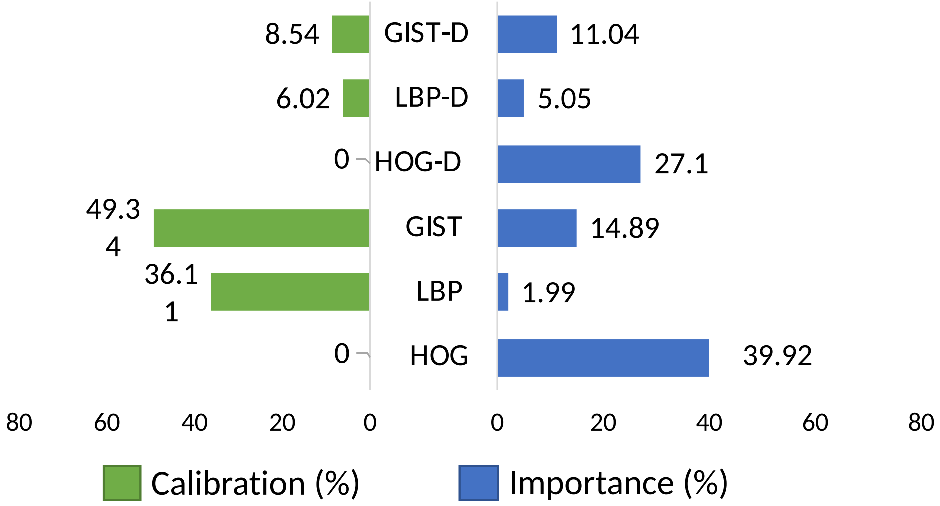
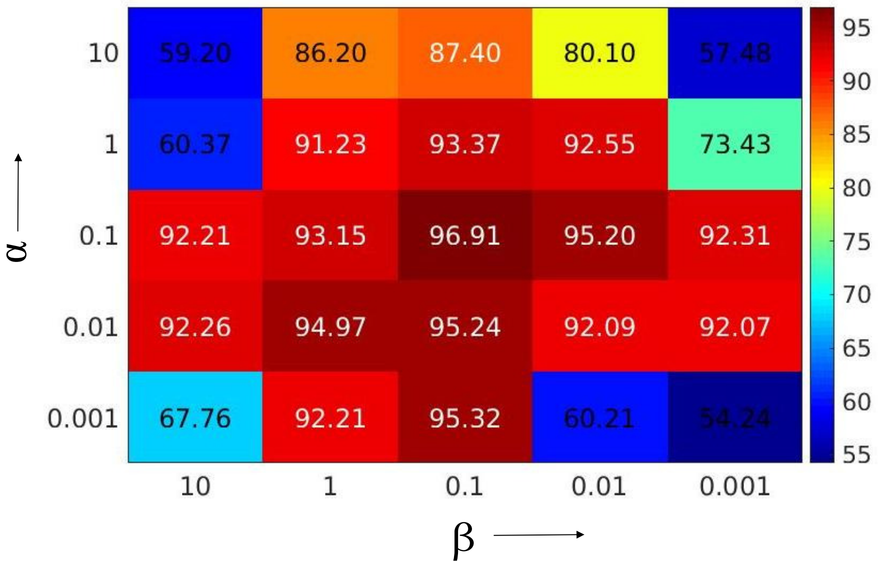


Figure 3.6 Experimental results on feature modality analysis during the training phase.

3.7 Discussion

3.7.1 High-Speed Processing

Due to ROPA's ability to integrate feature fusion, perception calibration, and classification in the unified formulation, and the efficiency of our convex objective function, our ROPA approach can achieve high-speed onboard processing. To validate this advantage, we perform additional experiments in Scenario I, using our CPU implementation on a Jackal robot's onboard computer. Without counting the time on extracting HOG, LBP, and GIST features, we obtain a processing rate at around 100 Hz. When counting the time for feature extraction, we obtain a processing rate of around 15 Hz. These results indicate the promise of ROPA to be applied in real-time robotics applications. In addition, any feature descriptor can be integrated to the ROPA approach, which provides the flexility to further improve the overall processing speed when faster, higher-quality features are available.



α \ β	10	1	0.1	0.01	0.001
10	59.20	86.20	87.40	80.10	57.48
1	60.37	91.23	93.37	92.55	73.43
0.1	92.21	93.15	96.91	95.20	92.31
0.01	92.26	94.97	95.24	92.09	92.07
0.001	67.76	92.21	95.32	60.21	54.24

Figure 3.7 Experimental results on hyperparameter analysis during the training phase.

3.7.2 Feature Modality Analysis

The proposed ROPA approach is capable to automatically estimate the importance of each feature modality, and how much each modality should be calibrated (i.e., calibration). We perform such experiments in the training phase using the data instances from Scenario I. The results are illustrated in Figure 3.6. The importance histogram indicates that, among the feature types used in the experiment, HOG is the most important in sensor fusion, followed by LBP, and GIST is the worst. The calibration graph illustrates that the GIST feature modality from the color data requires the most calibration, while the HOG modalities from both color and depth cues do not require calibration, in general. This result indicates that HOG features are relatively insensitive to the lighting changes.

3.7.3 Hyperparameter Selection

The hyperparameters α and β in our formulation, Eq. (3.4), are designed to control the strength of regularization terms over feature learning and perception calibration, respectively. Their optimal values can be determined using cross-validation during training. From the result in Figure 3.7, we observe that when $\alpha = \beta = 0.1$, ROPA statistically obtains the best accuracy. In general, the range of $\alpha, \beta \in (0.01, 1)$ can result in satisfactory accuracy, which also shows that both regularization terms are useful.

3.8 Summary

In this paper, we introduce the novel, bio-inspired approach named RObot Perceptual Adaptation (ROPA) to enable the new robot capability of calibrating multisensory perception, in order for robots to adapt to short-term and long-term environment changes. Our focused application in this paper aims at the task of long-term human following in the field environment, which is essential to real-world human-robot teaming applications.

The ROPA approach is formulated as a joint learning problem to simultaneously estimate the representativeness of each feature modalities, integrate heterogeneous features, and more importantly, calibrate the perception model to adapt multisensory perception with environmental changes. In order to fuse multisensory input data, we implement sparsity-inducing norms that enforces the base perception model to learn sparse weights of the multisensory input features and use the learned weights for multisensory fusion. To achieve perception calibration, we estimate the representativeness of the input features to encode the environment, and provide a calibration of the base model according to the environmental change. All components are integrated under the unified theoretical framework of regularized optimization.

In addition, we collect the new large-scale PEAC dataset containing multisensory data instances in scenarios of robot following of humans in a wide range of field environments with short-term and long-term environment changes. This open dataset provides a benchmark to evaluate and compare the approaches designed for robot perceptual adaptation to short-term and long-term variations of the robot operation environment in long-term human following applications. We conducted extensive experiments to evaluate and analyze ROPA in various scenarios using PEAC dataset. Experiment results have validated that, through calibrating perception, ROPA is able to effectively adapt to environment changes, obtains promising accuracy and efficiency, and outperforms baseline and previous methods in dynamic and unstructured environments in long-term human following applications.

The proposed mathematical formulation under regularized optimization for ROPA is general and has a potential to provide a framework that addresses robot adaptation to other environment and context changes (e.g., terrain changes and season changes). One of the challenges that prevent us from extending this approach to other adaptation scenarios is the lack of data on long-term robot adaptation, which will be our future work. Another future research can focus on integrating prior knowledge of cause and effect of environment changes in the adaptation process.

CHAPTER 4

ROBOT ADAPTATION TO UNSTRUCTURED TERRAINS BY JOINT REPRESENTATION AND APPRENTICESHIP LEARNING

This chapter is modified from a conference paper published at the 2019 Robotics: science and systems (RSS)[8]. This research was supported by NSF awards IIS-1849348, IIS-1849359, CNS-1823245, DOE grant DE-FE0031650, and ARO grant W911NF-17-1-0447.

Sriram Siva[9], Maggie Wigness[10] John Rogers[11] and Hao Zhang[12].

4.1 Abstract

When a mobile robot is deployed in a field environment, e.g., during a disaster response application, the capability of adapting its navigational behaviors to unstructured terrains is essential for effective and safe robot navigation. In this paper, we introduce a novel joint terrain representation and apprenticeship learning approach to implement robot adaptation to unstructured terrains. Different from conventional learning-based adaptation techniques, our approach provides a unified problem formulation that integrates representation and apprenticeship learning under a unified regularized optimization framework, instead of treating them as separate and independent procedures. Our approach also has the capability to automatically identify discriminative feature modalities, which can improve the robustness of robot adaptation. In addition, we implement a new optimization algorithm to solve the formulated problem, which provides a theoretical guarantee to converge to an optimal solution. In the experiments, we extensively evaluate the proposed approach in real-world scenarios, in which a mobile robot navigates on familiar and unfamiliar unstructured terrains. Experimental results have shown that the proposed approach is able to transfer human

[8] ©2019 RSS. Reprinted with permissions from Sriram Siva, Maggie Wigness, John Rogers, and Hao Zhang. ”Robot Adaptation to Unstructured Terrains by Joint Representation and Apprenticeship Learning”, in *Robotics: science and systems* (RSS), 2019.

[9] Primary researcher and author, Graduate Student, Colorado School of Mines

[10] Senior Research Scientist, United States DEVCOM Army Research Lab

[11] Senior Research Scientist, United States DEVCOM Army Research Lab

[12] Associate Professor, Colorado School of Mines

expertise to robots with small errors, achieve superior performance compared with previous and baseline methods, and provide intuitive insights on the importance of terrain feature modalities.

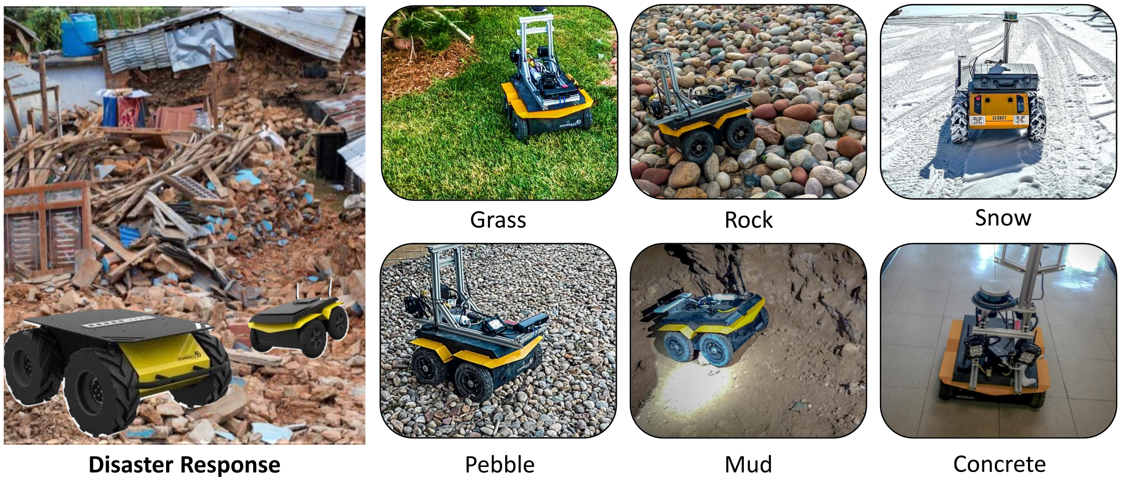


Figure 4.1 Motivating scenarios of robot adaptation to unstructured terrains. When mobile robots operates in a field environment during disaster response, the robots need the essential capability of navigating over a variety of terrains, such as grass, rock, snow, pebble, and mud, which often cannot be accurately structured before onsite robot deployment. In this paper, we propose the TRAL method to integrate representation learning and apprenticeship learning under a unified mathematical framework and estimate terrain feature importance to enable robust robot adaptation to unstructured terrains.

4.2 Introduction

Over the past several years, autonomous mobile robots have been more commonly used in unstructured field environments to address real-world applications, including disaster response, infrastructure inspection, homeland defense, and subterranean and planetary exploration [107, 108]. When deployed in an outdoor field environment, mobile robots need the essential capability of efficiently navigating through different types of terrains with a wide variety of characteristics. For example, as illustrated in Figure 4.1, when mobile robots perform disaster response tasks in a post-disaster situation, they are likely to encounter terrains such as grass, rock, pebble, mud, snow, and a mixture of these terrain types (e.g., a rocky terrain partially covered by mud and snow). In addition, the same type of terrains typically exhibits a variety of characteristics (e.g., different slope and softness). Such terrains whose types and characteristics cannot be fully modeled or determined before robot deployment are referred to as *unstructured terrains* [109, 110]. The capability of generating desired navigation behaviors (e.g., velocity control) that adapt to unstructured terrains is essential for an autonomous mobile robot to operate in complex field environments.

Given the importance of robot navigation over unstructured terrains, a large number of terrain classification and adaptation techniques were implemented [111–114].

Especially, learning-based techniques have been attracting increasing attention over the last few years. For example, unsupervised learning-based techniques were designed to perform semantic segmentation of terrains, and then navigational decisions were constructed by a robot through estimating traversability of unstructured terrains [115–117]. In addition, methods based upon terrain classification were implemented for mobile robots to reason about the possibility of traversing uneven terrains [118, 119]. Moreover, to teach navigational behaviors to robots, apprenticeship learning (a.k.a., learning from demonstration or imitation learning) was widely investigated [36, 109, 117] to transfer human expertise in navigation control to autonomous robots and generate adaptive behaviors according to given terrains (i.e., terrain adaptation).

Despite of promising performance of the previous methods, several challenges in robot adaptation to unstructured terrains have not yet been well addressed. Most existing learning-based methods focused on either terrain classification or navigation behavior generation only [120–124], or treated them as two separate components in the processing pipeline without a unified integration [36]. Moreover, previous adaptation techniques either used a single feature to represent terrains [125], or combined a number of features by simple concatenation [126]. The question of how to automatically estimate the importance of heterogeneous terrain features (e.g., from exteroperceptive and proprioceptive sensors) and fuse them together for robust adaptation has not been well answered.

In this paper, we propose a novel joint *terrain representation and apprenticeship learning* (TRAL) approach to enable robot adaptation to unstructured terrains. Instead of treating terrain classification and navigational behavior generation as separate tasks, our approach integrates them into a unified mathematical formulation under the regularized optimization paradigm. Our approach learns representations of unstructured terrains, which are jointly used to classify terrains and perform apprenticeship learning to generate navigational behaviors (e.g. velocity control) in the unified formulation. Furthermore, our approach is able to automatically learn weights of terrain features to create a multi-modal representation of terrains, by designing sparsity-induced norms as regularizers in the regularized optimization paradigm. These weights encode the importance of the terrain features, thus offering the insight of which terrain features are most critical for terrain classification and navigation behavior generation.

The key theoretical novelty of this chapter is twofold:

- We propose a novel TRAL method that is able to integrate representation learning and apprenticeship learning under a unified mathematical framework, automatically estimate terrain feature importance, and incorporate heterogeneous terrain features to enable robot adaptation to unstructured terrains.
- We develop a new optimization algorithm that efficiently solves the formulated joint representation and apprenticeship learning problem, which holds a theoretical guarantee to converge to an optimal solution.

The remaining of the chapter is organized as follows. Related work is reviewed in Section 8.2. We discuss the formulation of TRAL for robot adaptation to unstructured terrains in Section 8.3. The new optimization algorithm is described in Section 4.5. Experiments are presented in Section 4.6. Finally, we conclude the paper in Section 5.7.

4.3 Related Work

This section provides a review of related research on terrain classification and robot adaptation, and identifies key research challenges to be addressed in this paper.

4.3.1 Terrain Classification and Characterization

Terrain classification and characterization aims at extracting terrain features and associating them with terrain categories. Much of earlier methods was designed for larger vehicles and the characterization process was typically manual. There was an amount of research on Terramechanics [127], the guidance of autonomous vehicles through rough terrains. Similarly, Preplanning was used for speed selection [128], e.g., for vehicles used in the DARPA Grand Challenge, to address the problem of unstructured terrain navigation through fixing the optimal speed for each terrain type by collecting data for months. [129] generated digital terrain maps and then fitted a Gaussian model to classify terrains. [130] employed terrain cohesion and internal frictional angle obtained from a variety of terrains to perform terrain characterization.

In learning-based methods, terrain classification was usually performed by learning from terrain data in a regressive manner. For example, [131] designed a feed forward neural network on terrain images to classify terrain types. Motivated by the need to recognize visual terrain features that impact mobility, [132] used Hidden Markov Models to learn terrain features and an SVM classifier to identify terrain types. [120] used a series of sensors to measure robot vibrations on terrains to successfully classify terrains. [133] used a color-based classification to label obstacles based on terrain classes and carried out appropriate navigational behaviors.

Previous methods either used a single feature modality or a combination of features to represent terrains, and the problem of estimating the importance of feature modalities has not been well addressed yet, especially in the context of apprenticeship learning for robot terrain adaptation. Also, previous methods generally considered terrain classification and characterization as a preprocessing procedure for subsequential reasoning such as navigation planning.

4.3.2 Robot Adaptation

In the area of robot adaptation, high-level behavioral models were implemented to address the problem of robot adaptation in general [83, 134, 135]. Similarly, case-based reasoning [58] techniques were developed to make robots to adapt in dynamic environments [16, 17]. Many model-based approaches aimed at learning a single global model using function approximators, such as the Gaussian processes [136, 137] and neural networks [138]. Probabilistic Gaussian methods [139] were implemented to learn terrain models and update the models efficiently using sparse approximation. One of the key challenges of deploying such models is due to the difficulty of learning a global model that is accurate for the entire state space. Several approaches [140, 141] were developed to learn accurate local models instead of learning a global model. These techniques have acceptable local performance and allow for iterative local improvements of the policy, but they may not be feasible to be applied to an unexperienced domain. Recently, online learning approaches [66, 67] were also used to iteratively update model parameters during task execution. Although this online learning paradigm is widely used with robots operating in evolving environments, the effectiveness of the methods in the case of sudden change of terrain features is questionable.

For the research problem of robot adaptation to terrains, several approaches attempted to generate autonomously-adaptable dynamic maneuvering using a neural system model in simulations [142] and operational robots [143, 144]. Recently, methods based upon self-supervised learning [115, 145] were implemented to navigate in a minimal collision environment by maintaining minimal traversal and bumper history evaluations. [146] applied an array of inertial and ultrasonic sensors to calculate features of soil properties and carry out terrain adaptation. In the need for human-like operation and navigation [80], [117] used visual perception data and inverse optimal control trained with human supervision to learn to imitate expert navigation behaviors. By considering speed as an important factor for robot navigation, [147] presented a method for trading progress and velocity with respect to environment characteristics.

Given the promise of the previous learning-based techniques for robot adaptation to terrains, almost all existing approaches treated behavior generation and terrain classification as separate procedures, with no principled integration. Different from previous methods, our proposed approach explicitly addresses this challenge through integrating representation learning and apprenticeship learning under the same theoretical framework to perform joint terrain classification and behavior generation for terrain adaptation.

4.4 Approach

In this section, we introduce the novel TRAL approach that formulates robot adaptation to unstructured terrains as a joint representation and apprenticeship learning problem, from the unified regularized optimization perspective. An illustration of the TRAL approach is shown in Figure 4.2.

Notations: Matrices are denoted as boldface capital letters, and vectors as boldface lowercase letters. Given a matrix $\mathbf{M} = \{m_{ij}\} \in \mathbb{R}^{m \times n}$, we refer to its i-th row and j-th column as $\mathbf{m}^i$ and $\mathbf{m}_j$, respectively. The Frobenius norm of $\mathbf{M}$ is defined as $\|\mathbf{M}\|_F = \sqrt{\Sigma_{i=1}^m \Sigma_{j=1}^n m_{ij}^2}$. The ℓ_1-norm of a vector $\mathbf{v} \in \mathbb{R}^n$ is computed by $\|\mathbf{v}\|_1 = \Sigma_{i=1}^n |v_i|$. The ℓ_2-norm of $\mathbf{v}$ is computed by $\|\mathbf{v}\|_2 = \sqrt{\mathbf{v}^\top \mathbf{v}}$.

4.4.1 Representation Learning for Terrain Classification

We denote a collection of n data instances of terrain features that are acquired by a robot when traversing a terrain as $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$, where $\mathbf{x}_i \in \mathbb{R}^d$ denotes a feature vector of length d. We represent terrain types associated with $\mathbf{X}$ as $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_n] \in \mathbb{R}^{c \times n}$, where $\mathbf{z}_i \in \mathbb{Z}^c$ is an indicator vector with each element $z_{ij} \in \{0, 1\}$ indicating whether the i-th data instance of features $\mathbf{x}_i$ has the j-th terrain type, and c is the number of terrain types. Given $\mathbf{X}$ and $\mathbf{Z}$, representation learning for terrain classification can be simply formulated as an optimization problem [47, 148]:

$$\min_{\mathbf{V}} \|\mathbf{V}^\top \mathbf{X} - \mathbf{Z}\|_F^2 \tag{4.1}$$

where $\mathbf{V} \in \mathbb{R}^{d \times c}$ is a weight matrix with v_{ij} representing the importance of the i-th element in a feature vector with respect to the j-th terrain type. The loss function parameterized by $\mathbf{V}$ in Eq. (4.1) models the squared error of utilizing the weighted features to represent the terrain types.

When the terrain features are computed by different feature extraction techniques and/or from different sensors, the feature vector contains multiple feature modalities, with each modality including features computed by one technique from one sensor. Then, each feature vector $\mathbf{x}_i$ exhibits a multi-modal structure $\mathbf{x}_i = [(\mathbf{x}_i^1)^\top, \dots, (\mathbf{x}_i^m)^\top]^\top \in \mathbb{R}^d$, where m is the number of feature modalities in the vector. If d_j is the length of the j-th feature modality, we have $d = \Sigma_{j=1}^m d_j$.

Because different terrain features typically capture different characteristics of unstructured terrains (e.g., color, smoothness, and roughness) and often contribute differently toward terrain classification, it is essential to identify most descriptive feature modalities when multiple modalities are used together. In order to enable this capability, we develop a norm to regularize the weight matrix $\mathbf{V}$ [54]:

$$\|\mathbf{V}\|_R = \sum_{i=1}^{c} \sum_{j=1}^{m} \|\mathbf{v}_i^j\|_2 = \sum_{i=1}^{c} \sum_{j=1}^{m} \sqrt{\sum_{k=1}^{d_j} \left(v_{ik}^j\right)^2} \tag{4.2}$$

where $\mathbf{v}_i^j \in \mathbb{R}^{d_i}$ represents the weight vector of terrain features in the i-th modality with respect to the j-th terrain type. Since $\|\mathbf{V}\|_R$ uses the ℓ_2-norm within each modality and the ℓ_1-norm among modalities with respect to each terrain type, it enforces sparsity between modalities to identify the most discriminative feature modalities. That is, if a modality is not discriminative to represent the terrains for terrain classification, the weights of terrain features in this modality are assigned with zeros (in ideal case, usually they are small values close to 0); otherwise, the weights have big values. The effect of the norm $\|\mathbf{V}\|_R$ is illustrated in Figure 4.2.

Then, representation learning from multi-modal features for terrain classification is formulated as the following regularized optimization problem:

$$\min_{\mathbf{V}} \|\mathbf{V}^\top \mathbf{X} - \mathbf{Z}\|_F^2 + \lambda_1 \|\mathbf{V}\|_R \tag{4.3}$$

where $\lambda_1 \geq 0$ is a trade-off hyper-parameter applied to balance the loss function and the regularization term.

4.4.2 Formulation of Joint Representation and Apprenticeship Learning for Unstructured Terrain Adaptation

Apprenticeship learning [80] is one of the most widely used learning-based paradigms to transfer human expertise to robots through directly expert demonstration, which can be viewed as the learning of a projection from observation or feature space into behavior space, e.g., by probabilistic [149] or optimization-based [36, 150] approaches.

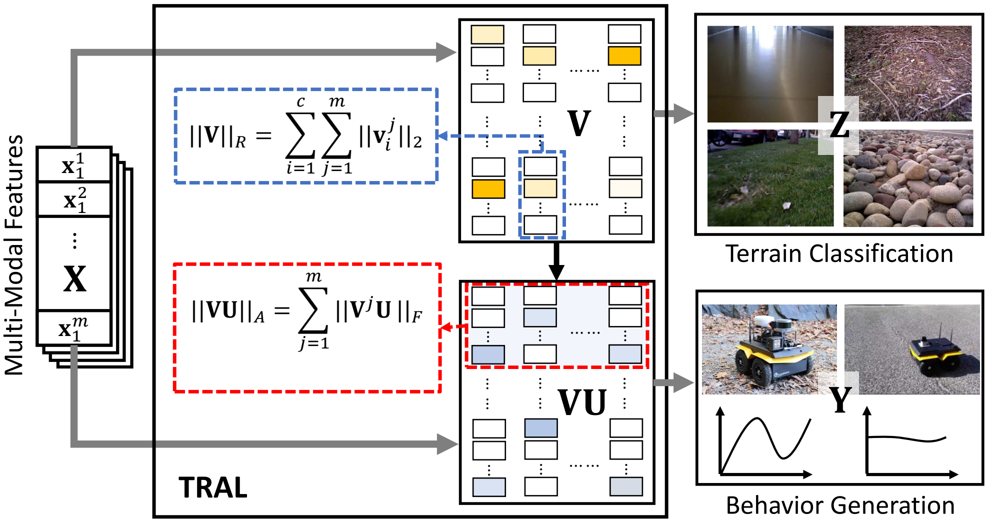

Multi-Modal Features
$\mathbf{x}_1^1$
$\mathbf{x}_1^2$
$\mathbf{X}$
$\mathbf{x}_1^m$
$||\mathbf{V}||_R = \sum_{i=1}^{c}\sum_{j=1}^{m}||\mathbf{v}_i^j||_2$
$||\mathbf{VU}||_A = \sum_{j=1}^{m}||\mathbf{V}^j\mathbf{U}||_F$
V
VU
TRAL
Z
Terrain Classification
Y
Behavior Generation

Figure 4.2 Illustration of the proposed TRAL approach for joint representation and apprenticeship learning under the unified regularized optimization framework, with the regularization terms $\|\mathbf{V}\|_R$ and $\|\mathbf{VU}\|_A$ to identify the discriminative feature modalities.

Our goal in this paper is to achieve apprenticeship learning while retaining the advantage of discriminative features offered by representation learning. The theoretical novelty of the paper is that we introduce a principled integration of representation and apprenticeship learning in a unified formulation under the mathematical framework of regularized optimization, which enables joint terrain classification and behavior generation.

Formally, given expert demonstrations on behavior controls $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n] \in \mathbb{R}^{r \times n}$ (e.g., velocity, motor torque, and consumed power) that are associated with terrain features $\mathbf{X}$ during robot navigation, where r is the number of controls we want to learn for behavior generation, we formulate joint representation and apprenticeship learning for terrain adaptation as a unified regularized optimization problem:

$$\min_{\mathbf{V},\mathbf{U}} \quad \|\mathbf{V}^\top \mathbf{X} - \mathbf{Z}\|_F^2 + \|(\mathbf{VU})^\top \mathbf{X} - \mathbf{Y}\|_F^2 + \lambda_1 \|\mathbf{V}\|_C + \lambda_2 \|\mathbf{VU}\|_A \qquad (4.4)$$

where λ_2 is a tradeoff hyper-parameter. The new loss function (i.e., the 2nd term in Eq. (4.4)) encodes the difference between the learning model and expert demonstrations, in order to learn the projection from multi-modal feature space to robot behavior space. This loss function is parameterized by $\mathbf{VU} \in \mathbb{R}^{d\times r}$, which first projects input multi-modal terrain features $\mathbf{X}$ into a more discriminative space $\mathbf{V}^\top\mathbf{X}$ (parameterized by $\mathbf{V}$ from representation learning), and further projects the representation space into the robot behavior space $\mathbf{U}^\top\mathbf{V}^\top\mathbf{X}$ (parameterized by $\mathbf{U} \in \mathbb{R}^{c\times r}$ for apprenticeship learning).

Because the output of the proposed joint learning approach is to generate robot behaviors, we also want our representation learning to learn discriminative terrain features that can better decide robot navigation behaviors. Therefore, we design a new norm for $\mathbf{VU}$, which is mathematically defined as:

$$\|\mathbf{VU}\|_A = \sum_{j=1}^{m} \|\mathbf{V}^j\mathbf{U}\|_F \tag{4.5}$$

where $\mathbf{V}^j \in \mathbb{R}^{d_j\times c}$ is the block of $\mathbf{V}$ that includes weights of terrain features belonging to the j-th modality. This norm can be directly integrated into our objective function in Eq. (4.4) as a regularization term over $\mathbf{VU}$, as demonstrated in Figure 4.2. This regularization enforces sparsity among feature modalities with respect to all robot behaviors, thus facilitating identifying discriminative features for robot behavior generation.

After solving the formulated regularized optimization problem in Eq. (4.4) based on Algorithm 3, we can obtain the optimal $\mathbf{V}^*$ and $\mathbf{U}^*$. When a new multi-modal terrain feature vector $\mathbf{x}^o \in \mathbb{R}^d$ is acquired by a robot during execution, the robot's navigation behaviors can be determined by:

$$\mathbf{y}^o = (\mathbf{V}^*\mathbf{U}^*)^\top\mathbf{x}^o \tag{4.6}$$

which incorporates considerations of both representation learning and apprenticeship learning.

Different from conventional learning-based robot adaptation techniques [36, 117, 118], TRAL provides a unified formulation that integrates representation and apprenticeship learning under the same theoretical framework, instead of treating them as separate and independent procedures. Moreover, different from most existing apprenticeship learning methods that apply a single feature modality only [150] or simple feature concatenation [36], one of the TRAL's advantages is its

Algorithm 3: An iterative algorithm to solve the formulated regularized optimization problem in Eq. (4.4)

```
Input  : $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$
Let $t = 1$. Initialize $\mathbf{V}(t)$ by solving $\min_{\mathbf{V}} \|\mathbf{V}^\top\mathbf{X} - \mathbf{Z}\|_F^2$
Initialize $\mathbf{U}(t)$ by solving $\min_{\mathbf{U}} \|(\mathbf{V}\mathbf{U})^\top\mathbf{X} - \mathbf{Y}\|_F^2$
while not converge do
    Calculate the block diagonal matrix $\mathbf{D}(t+1)$, where the $i$-th diagonal element of
      $\mathbf{D}(t+1)$ is $\frac{\mathbf{1}}{2\|\mathbf{v}_i(t)\|_2}\mathbf{I}_i$
    Calculate the block diagonal matrix $\widehat{\mathbf{D}}(t+1)$, where the $j$-th diagonal element is
      $\frac{1}{2\|\mathbf{V}^j\mathbf{U}(t)\|_F}\mathbf{I}_j$
    For each $\mathbf{v}_i(1 \le i \le c) = (\mathbf{X}\mathbf{X}^\top + \lambda_1\mathbf{D} + \lambda_2\widehat{\mathbf{D}})^{-1}\mathbf{X}(\mathbf{z}_i)^\top$
    For each $\mathbf{u}_i(1 \le i \le c) = (\mathbf{V}^\top\mathbf{X}\mathbf{X}^\top \mathbf{V} + \lambda_2\mathbf{V}^\top\mathbf{V})^{-1}\mathbf{X}\mathbf{y}_i$
    $t = t + 1$
Output : $\mathbf{V} = \mathbf{V}(t) \in \Re^{d\times c}$ and $\mathbf{U} = \mathbf{U}(t) \in \Re^{c\times r}$
```

capability to automatically identify discriminative feature modalities, which can improve the robustness of robot adaptation to unstructured and unfamiliar terrains. Also, the weights learned by TRAL provide insights for intuitive analysis and future integration of heterogeneous terrain features for robot adaptation to unstructured terrains.

4.5 Optimization Algorithm

The second theoretical novelty of this paper is the design of an optimization algorithm to solve the formulated regularized optimization problem in Eq. (4.4). Although the problem in Eq. (4.4) is convex, it is challenging to solve using traditional gradient descent based solvers due to the non-smooth regularization terms and due to the dependency of the two loss functions. Our optimization algorithm follows the iterative re-weighted method [55].

To solve the formulated problem, we compute the derivative of Eq. (4.4) with respect to the columns of $\mathbf{V}$ and set it to zero, while assuming $\mathbf{U}$ to have a fixed value:

$$\mathbf{X}\mathbf{X}^\top\mathbf{v}_i - \mathbf{X}(\mathbf{z}_i)^\top + \lambda_1\mathbf{D}\mathbf{v}_i + \lambda_2\widehat{\mathbf{D}}\mathbf{v}_i = 0 \tag{4.7}$$

where $\mathbf{D}$ is a diagonal matrix with the i-th diagonal element as $\frac{\mathbf{I}}{2\|\mathbf{v}_i\|_2}$, and $\widehat{\mathbf{D}}$ represents a block diagonal matrix with the j-th block computed by $\frac{\mathbf{I}}{2\|\mathbf{V}^j\mathbf{U}\|_F}$. Then, we obtain:

$$\mathbf{v}_i = (\mathbf{X}\mathbf{X}^\top + \lambda_1\mathbf{D} + \lambda_2\widehat{\mathbf{D}})^{-1}\mathbf{X}(\mathbf{z}_i)^\top \tag{4.8}$$

Fixing the value of $\mathbf{V}$, to compute $\mathbf{U}$, we take the derivative of Eq. (4.4) with respect to the columns of $\mathbf{U}$ and set it to zero:

$$\mathbf{V}^\top\mathbf{X}\mathbf{X}^\top\ \mathbf{V}\mathbf{u}_i - \mathbf{V}^\top\mathbf{X}(\mathbf{y}_i)^\top + \lambda_2\mathbf{V}^\top\mathbf{V}\mathbf{u}_i = 0 \tag{4.9}$$

Then we obtain:

$$\mathbf{u}_i = (\mathbf{V}^\top\mathbf{X}\mathbf{X}^\top\ \mathbf{V} + \lambda_2\mathbf{V}^\top\mathbf{V})^{-1}\mathbf{X}\mathbf{y}_i \tag{4.10}$$

Since both $\mathbf{D}$ and $\widehat{\mathbf{D}}$ are dependent on $\mathbf{V}$, they are unknown parameters that we need to estimate. Accordingly, we develop an iterative algorithm to solve the formulated problem, which is described in Algorithm 3.

As follows, we provide a theoretical analysis of Algorithm 3. We first show a lemma cited from Nie et al. [151]:

Lemma 3: For any two given vectors $\mathbf{v}$ and $\tilde{\mathbf{v}}$, the following inequality relation holds:

$$\|\tilde{\mathbf{v}}\|_2 - \frac{\|\tilde{\mathbf{v}}\|_2^2}{2\|\mathbf{v}\|_2} \leq \|\mathbf{v}\|_2 - \frac{\|\mathbf{v}\|_2^2}{2\|\mathbf{v}\|_2}$$

From Lemma 6, we can derive the following corollary:

Corollary 2: For given matrices $\mathbf{M}$ and $\tilde{\mathbf{M}}$, the following inequality relation holds:

$$\|\tilde{\mathbf{M}}\|_F - \frac{\|\tilde{\mathbf{M}}\|_F^2}{2\|\mathbf{M}\|_F} \leq \|\mathbf{M}\|_F - \frac{\|\mathbf{M}\|_F^2}{2\|\mathbf{M}\|_F}$$

Then, we prove our algorithm's convergence as follows:

Theorem 3: Algorithm 3 iteratively decreases the value of the objective function in Eq. (4.4) and converges to a local optimal solution.

Proof. According to Step 4 of Algorithm 3, we obtain:

$$\mathbf{V}(t+1) = \min_{\mathbf{V}} \|\mathbf{V}^\top \mathbf{X} - \mathbf{Z}\|_F^2 + \lambda_1 Tr \mathbf{V}^\top \mathbf{D}(t+1)\mathbf{V} + \lambda_2 Tr \mathbf{V}^\top \widehat{\mathbf{D}}(t+1)\mathbf{V} \tag{4.11}$$

Also from Step 5 of Algorithm 3, we obtain:

$$\mathbf{U}(t+1) = \min_{\mathbf{U}} \|(\mathbf{V}\mathbf{U})^\top \mathbf{X} - \mathbf{Y}\|_F^2 + \lambda_2 Tr \mathbf{U}^\top \mathbf{V}^\top(t+1)\mathbf{V}(t+1)\mathbf{U} \tag{4.12}$$

Then, we can derive that

$$\begin{aligned} \mathcal{J}(t+1) + \mathcal{F}(t+1) + \lambda_1 Tr \mathbf{V}^\top(t+1)\mathbf{D}(t+1)\mathbf{V}(t+1) \\ + \lambda_2 Tr \mathbf{U}^\top(t+1)\widehat{\mathbf{D}}(t+1)\mathbf{U}(t+1) \\ \leq \mathcal{J}(t) + \mathcal{F}(t)\lambda_1 Tr \mathbf{V}^\top(t)\mathbf{D}(t+1)\mathbf{V}(t) + \lambda_2 Tr \mathbf{U}^\top(t)\widehat{\mathbf{D}}(t+1)\mathbf{U}(t) \end{aligned} \tag{4.13}$$

where $\mathcal{J}(t)$ is given as $\|\mathbf{V}^\top \mathbf{X}(t) - \mathbf{Z}\|_F^2$ and $\mathcal{F}(t)$ as
$\|(\mathbf{V}\mathbf{U})^\top \mathbf{X}(t) - \mathbf{Y}\|_F^2 + \lambda_2 Tr \mathbf{U}^\top \mathbf{V}^\top(t)\mathbf{V}(t)\mathbf{U}$.
After substituting the definition of $\mathbf{D}$ and $\widehat{\mathbf{D}}$, we obtain

$$\begin{aligned} \mathcal{J}(t+1) + \mathcal{F}(t+1) + \lambda_1 \sum_{i=1}^{c}\sum_{j=1}^{m} \frac{\|\mathbf{v}_i^j(t+1)\|_2^2}{2\|\mathbf{v}_i^j(t)\|_2} + \lambda_2 \frac{\|\mathbf{V}(t+1)\mathbf{U}(t+1)\|_F^2}{2\|\mathbf{V}(t)\mathbf{U}(t)\|_F} \\ \leq \mathcal{J}(t) + \mathcal{F}(t) + \lambda_1 \sum_{i=1}^{c}\sum_{j=1}^{m} \frac{\|\mathbf{v}_i^j(t)\|_2^2}{2\|\mathbf{v}_i^j(t)\|_2} + \lambda_2 \frac{\|\mathbf{V}(t)\mathbf{U}(t)\|_F^2}{2\|\mathbf{V}(t)\mathbf{U}(t)\|_F} \end{aligned} \tag{4.14}$$

From Lemma 6, we can derive:

$$\begin{aligned} \sum_{i=1}^{c}\sum_{j=1}^{m} \|\mathbf{v}_i^j(t+1)\|_2 - \sum_{i=1}^{c}\sum_{j=1}^{m} \frac{\|\mathbf{v}_i^j(t+1)\|_2^2}{2\|\mathbf{v}_i^j(t)\|_2} \\ \leq \sum_{i=1}^{c}\sum_{j=1}^{m} \|\mathbf{v}_i^j(t)\|_2 - \sum_{i=1}^{c}\sum_{j=1}^{m} \frac{\|\mathbf{v}_i^j(t)\|_2^2}{2\|\mathbf{v}_i^j(t)\|_2} \end{aligned} \tag{4.15}$$

and from Corollary 4, we can derive:

$$\|\mathbf{V}(t+1)\mathbf{U}(t+1)\|_F - \frac{\|\mathbf{V}(t+1)\mathbf{U}(t+1)\|_F^2}{2\|\mathbf{V}(t)\mathbf{U}(t)\|_F} \leq \|\mathbf{V}(t)\mathbf{U}(t)\|_F - \frac{\|\mathbf{V}(t)\mathbf{U}(t)\|_F^2}{2\|\mathbf{V}(t)\mathbf{U}(t)\|_F} \quad (4.16)$$

Adding Eq. (7.14)-(4.16) on both sides, we obtain:

$$\begin{aligned} &\mathcal{J}(t+1) + \mathcal{F}(t+1) + \lambda_1 \sum_{i=1}^{c}\sum_{j=1}^{m} \|\mathbf{v}_i^j(t+1)\|_2 + \lambda_2 \|\mathbf{V}(t+1)\mathbf{U}(t+1)\|_A \quad (4.17) \\ &\leq \mathcal{J}(t) + \mathcal{F}(t) + \lambda_1 \sum_{i=1}^{c}\sum_{j=1}^{m} \|\mathbf{v}_i^j(t)\|_2 + \lambda_2 \|\mathbf{V}(t)\mathbf{U}(t)\|_A \end{aligned}$$

Eq. (7.20) decreases the value of the objective function in each iteration. Because our objective function is convex, Algorithm 3 converges to a local optimal solution of the formulated regularized optimization problem in Eq. (4.4). □

Complexity analysis: Since the objective function in Eq. (4.4) is convex, Algorithm 3 converges very fast. In each iteration of Algorithm 3, it is trivial to compute Steps 4 and 5. We can compute Steps 6 and 7 by solving a system of linear equations, with a quadratic complexity.

4.6 Experiments

This section discusses the experiment setup, present implementation details, and provides an analysis of our experimental results and comparison with other methods.

4.6.1 Experiment Setup and Dataset

In the experiments, a Clearpath Jackal mobile robot is used to navigate over a variety of unstructured terrains. In order to collect heterogeneous multi-sensory data that can be utilized to improve terrain representations, the mobile robot is equipped with multiple diverse proprioceptive and exteroceptive sensors. A structured-light camera is applied as an exteroceptive sensor to capture 3D colored point clouds of unstructured terrains in front of the robot with a frame rate of 30 Hz. Proprioceptive sensors are also installed on the robot to measure its internal information, including wheel odometry, motor speed, inertial measurement unit (IMU) measurement, power consumption, and battery status. The data frame rate from the proprioceptive sensors varied from 5–100 Hz. Linear interpolation based on data time-stamps is performed to ensure data frame rate to be consistent at 30 Hz. All of these data sources are used together as the input to terrain adaptation approaches for evaluation and comparison.

In addition, expert controls of the mobile robot are recorded as demonstrations of robot behaviors with the goal to navigate the mobile robot over unstructured terrains in straight lines as quickly as possible while maintaining safety (e.g., no flipping or crushing). The recorded demonstrations are utilized as the input to apprenticeship learning-based methods for training.

In the experiments, five types of unstructured terrains with various characteristics are used (shown in Figure 4.3), including:

- Concrete: The concrete surface exhibits smooth and hard characteristics. The mobile robot navigates over concrete floors in an indoor corridor.

- Grass: The grass terrain is less smooth compared to the concrete terrain, and is characterized by low ruggedness. The grass terrain often causes small wheel slips when the mobile robot navigates in places with a greater slope.

- Mud: This type of unstructured terrains has special characteristics in terms of roughness. Muddy ground surface is wet and typically soft, and can produce a lot of slips in robot motion. Mud terrains may also consist of weathered leaves and wood chips.

- Pebble: The pebble terrain is rough and characterized by small rocks, which we refer to as pebbles. The roughness requires a robot to navigate slowly and also results in relatively high power consumption. During demonstration, the robot is operated by human experts in a way to avoid sudden jerks, which is desirable in scenarios when a robot operates with fragile equipment.

- Rock: The rock terrain consists of larger rocks. This is the most challenging terrain type used for robot navigation in our experiments. The experts need to carefully control the robot and make detours to ensure safety of the mobile robot.

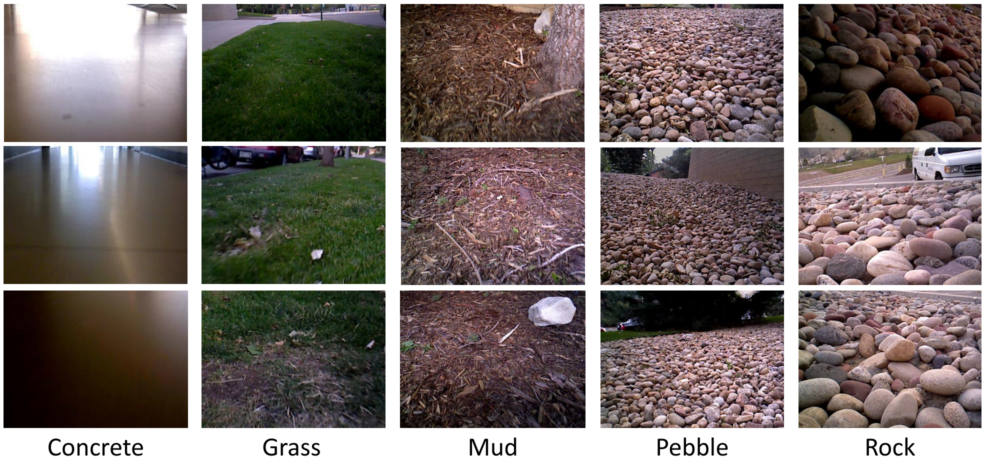


Figure 4.3 Instances of the five terrain types used in the experiments. The color images are observed by the camera installed in front of the mobile robot when navigating over these unstructured terrains.

Experiments of robot navigation are performed twenty times on each of the five unstructured terrain types. Each experiment lasts for a period of 10-25 seconds of an expert operating the mobile robot to navigate over a particular terrain type. In total, the experiments consist of 4000 data instances. All instances are organized into a dataset, which contains the heterogeneous multi-sensory data obtained by the robot during navigation on unstructured terrains, as well as the associated demonstrations.

4.6.2 Implementation

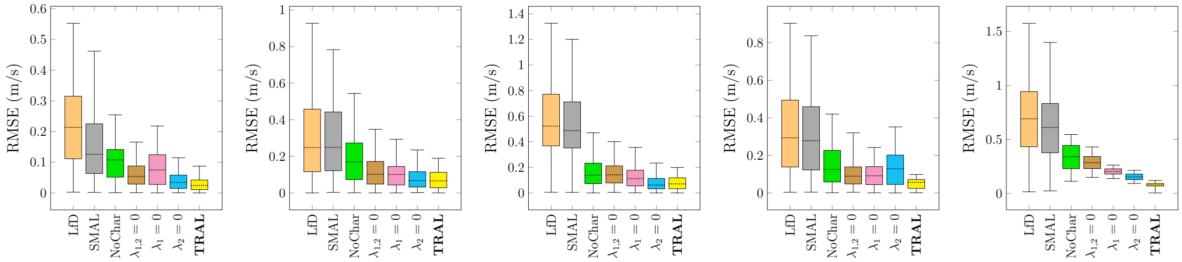
From the multi-sensory data obtained by robot sensors, commonly adopted terrain features are extracted. For exteroceptive data from depth and color images, real-time feature extraction techniques, including Histogram of Oriented Gradients (HOG) [101], color descriptors, and Local Binary Patterns (LBP) [152], are used to compute visual features of the terrains from both color and depth images. The extracted HOG, Color and LBP features capture different visual characteristics of the terrains. Additionally, the point cloud data collected by the color-depth sensor is used to compute Elevation Maps based on grids [153] to discretely represent the grid-wise elevation and variance of the unstructured terrain. 3D point clouds with transformation and wheel odometry data are utilized to form the grid-based local elevation map. Normal vector maps and variance maps are also calculated based upon the generated elevation map of unstructured terrains. The proprioceptive sensor data including IMU and wheel odometry, are used together with the visual and elevation features to form a feature vector that is used as the input to our TRAL approach.

We implement four versions of the TRAL approach. First, we set λ_1 and λ_2 in Eq. (4.4) to 0, which performs joint terrain classification and robot behavior generation but cannot identify discriminative features. Then, we set $\lambda_1 = 0$ and $\lambda_2 = 0.1$ to identify discriminative features only useful for terrain classification. In addition, we assign $\lambda_1 = 0.1$ and $\lambda_2 = 0$ to identify discriminative features only useful for behavior generation. Finally, the full version of the proposed TRAL approach is implemented with $\lambda_1 = \lambda_2 = 0.1$, which identifies features that contribute to the joint terrain classification and behavior generation. We observe that TRAL shows stable performance when $\lambda_1, \lambda_2 \in (0.05, 5)$. The TRAL approach is implemented in Linux on the Jackal robot's onboard computer with an i5 2.5 GHz CPU and 8G memory.

We also compare the TRAL appraoch with previous apprenticeship learning methods used for robot adaptation, including Learning from Demonstration (LfD) based upon probabilistic graphical models [154], Sequence-based Multi-modal Apprenticeship Learning (SMAL) [36] that implements robot adaptation based upon optimization-based state perception and MDP-based robot decision making, and an imitation learning method using optimization-based multiclass classification [150] with no representation learning (NoChar).

4.6.3 Result on Familiar Terrains

In this set of experiments, we evaluate the proposed TRAL approach's performance when the mobile robot navigates over familiar unstructured terrains, in which the terrain types have been experienced by the robot during the training period. The robot is trained using terrain features and expert demonstrations collected in all five types of unstructured terrains used, and then tested using terrain features only from the five types of terrains. In order to evaluate how well human expertise is transferred to a mobile robot in terms of ground speed, which is viewed as one of the most critical performance metrics that indicates how fast a navigation task can be completed by the robot [128, 155], we use the Root Mean Square Error (RMSE) as the metric to measure the magnitude of deviation of navigation behaviors generated by apprenticeship learning methods from the expert controls. Although RMSE provides the magnitude of deviation, it cannot present the ratio of deviation, comparing with the expert control. Therefore, we also use error ratio (ER), defined as the average of the ratio of the deviation magnitude over the control magnitude, in order to measure the percentage of deviation from the expert controls. For both RMSE and ER, a smaller value indicates a better performance.



(a) Concrete (b) Grass (c) Mud (d) Pebble (e) Rock

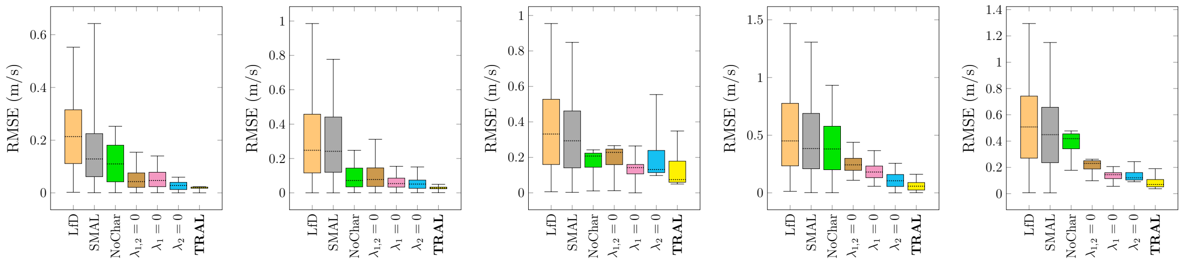
Figure 4.4 Quantitative results of RMSE on ground velocity obtained by TRAL in the scenarios of navigation over *familiar* unstructured terrains. Comparison with previous apprenticeship learning methods and our baseline methods is also presented. The results of each method over each type of terrains are presented as a boxplot that depicts the distribution of the results from all testing instances based on a five number summary (i.e., minimum, first quartile, median, third quartile, and maximum).

The quantitative result obtained by our proposed approach is demonstrated in Figure 4.4 in the scenario of navigation over each of the familiar unstructured terrains, using a boxplot that displays the distribution of the results from all testing instances based on a five number summary (i.e., minimum, first quartile, median, third quartile, and maximum). The overall mean and variance of RMSE and ER on ground velocity obtained by our TRAL approach are summarized in Table Table 4.2. TRAL achieves the RMSE of 0.05 ± 0.03 m/s, which shows the magnitude of deviation is 5 cm/s from expert control on average. The error rate ER demonstrates that the difference between the generated control and expert control is 2.41% with a variance of 1.67% for robots navigating over familiar terrains.

Table 4.1 Mean and variance of RMSE and ER on ground velocity obtained by TRAL and previous/baseline methods over familiar unstructured terrains.

Approach	RMSE (m/s)	ER (%)
LfD [154]	0.39 (±0.20)	21.35 (±10.16)
SMAL [36]	0.34 (±0.18)	12.5 (±10.10)
NoChar [150]	0.17 (±0.08)	10.75 (±4.12)
$\lambda_{1,2} = 0$	0.13 (±0.05)	7.48 (±4.96)
$\lambda_1 = 0$	0.11 (±0.04)	5.33 (±3.40)
$\lambda_2 = 0$	0.08 (±0.04)	3.40 (±2.41)
TRAL	0.05 (±0.03)	2.41 (±1.76)

Quantitative comparisons with several previous approaches and baseline methods are demonstrated in Figure 4.4 using the boxplot graph and comparisons of the overall performance are summarized in Table Table 4.1. It is observed that the previous apprenticeship learning methods tested in our experiment, including LfD, SMAL, and NoChar obtain a larger deviation magnitude and percentage. A likely reason that explains this phenomenon is that they cannot perform representation learning to identify discriminative terrain features. The raw feature vectors directly computed from the sensor measurements are noisy, which can decrease the performance of navigational behavior generation. Since our approach combines two loss functions to perform a joint optimization for both representation and apprenticeship learning, even without applying the two regularization norms, the version of our approach with $\lambda_{1,2} = 0$ is able to build good representations to improve the performance of behavior generation. When regularization terms are used to promote modality sparsity (i.e., with non-zero λ_1 or λ_2), these versions of our approach can find more discriminative features toward terrain classification or behavior generation, which can further reduce the deviation. When both regularization norms are used to learn terrain representations toward both terrain classification and navigational behavior generation, the complete TRAL approach obtains superior performance on both RMSE and ER over the previous methods.



(a) Concrete (b) Grass (c) Mud (d) Pebble (e) Rock

Figure 4.5 Quantitative results of RMSE on ground velocity obtained by TRAL in the scenarios of navigation over *unfamiliar* unstructured terrains. Comparison with previous apprenticeship learning methods and our baseline methods is also presented.

4.6.4 Result on Unfamiliar Unstructured Terrains

In this set of experiments, we evaluate TRAL's performance when the mobile robot navigates over unfamiliar unstructured terrains, in which the terrain type that the robot navigates over has not been experienced during training. The robot is trained using terrain features and expert demonstrations collected in four categories of the unstructured terrains, and then evaluated using the fifth terrain type that has not been used in the training period. For example, Figure 4.5(c) demonstrates the results from the scenario that the robot is trained using data from concrete, grass, pebble and rock terrain types, and evaluated on the mud terrain what is not experienced by the robot in training.

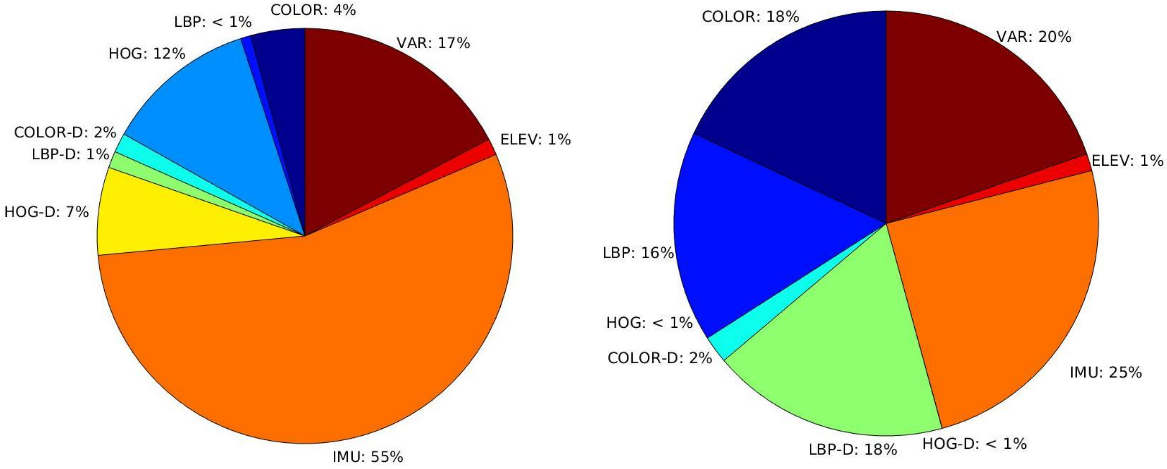
The RMSE results obtained by TRAL and comparisons with previous and baseline techniques on each type of terrains that are not familiar to the robot are demonstrated in Figure 4.5. The overall performance of TRAL and other methods are presented in Table Table 4.2. When the robot navigates over unfamiliar terrains, we observe phenomena that are similar to the results obtained when it navigates on familiar terrains: apprenticeship learning approaches without using representation learning do not work well; integrating the representation and apprenticeship learning decreases the errors; and the proposed regularization terms can further increase the performance. The complete TRAL method still obtains the best performance over other methods, mostly because of its capability of identifying the most discriminative terrain features that can generalize well to unfamiliar scenarios. On the other hand, since unfamiliar terrains are not seen in the training period, all approaches evaluated in this case exhibit higher errors (i.e., higher RMSE and ER values) comparing to the results from the situations when robots navigate on familiar unstructured terrains.

Table 4.2 Mean and variance of RMSE and ER on ground velocity obtained by TRAL and previous/baseline methods over unfamiliar unstructured terrains.

Approach	RMSE (m/s)	ER (%)
LfD [154]	0.35 (±0.21)	33.15 (±19.59)
SMAL [36]	0.29 (±0.19)	29.35 (±16.80)
NoChar [150]	0.23 (±0.07)	20.73 (±6.26)
$\lambda_{1,2} = 0$	0.16 (±0.05)	22.82 (±6.97)
$\lambda_1 = 0$	0.11 (±0.03)	14.22 (±7.52)
$\lambda_2 = 0$	0.08 (±0.04)	13.14 (±4.56)
TRAL	0.06 (±0.03)	7.45 (±3.89)

4.6.5 Discussion

The TRAL approach offers the desirable ability to automatically estimate the importance of feature modalities for terrain classification and behavior generation. The results of modality importance averaged from both familiar and unfamiliar experimental scenarios are demonstrated in Figure 4.6. It is observed that two feature modalities, i.e., IMU and terrain variance, are most discriminative in determining both terrain types (from Figure 4.6(a)) and robot navigational behaviors (from Figure 4.6(b)). Intuitively, both feature modalities can be considered to contain complementary information because the IMU data is greatly affected by and correlated with terrain variances. Moreover, it is interesting to note that HOG features are discriminative for classifying the terrain types, but not for learning robot navigation behaviors. LBP features on the other hand exhibit the opposite impact. When comparing all visual features, we observe that features extracted from color images are generally more discriminative than the features from depth images. Due to TRAL's ability to integrate representation and apprenticeship learning in the unified framework, and due to the efficiency of our convex optimization formulation, TRAL is able to achieve high-speed processing on the robot's onboard computer. Including the time spent on feature extraction, our approach obtains a processing rate of 20 Hz. Without feature extraction, our approach itself achieves a processing speed of 80 Hz on average. This result indicates the runtime advantage of our TRAL approach for real-time robotics applications and its potential to facilitate high-speed mobile robot navigation.



(a) Terrain classification

(b) Behavior generation

Figure 4.6 Normalized importance of terrain feature modalities with respect to terrain classification (Figure 4.6(a)) and behavior generation (Figure 4.6(b)).

4.7 Summary

In this paper, we present the TRAL approach to address the research problem of robot adaptation to unstructured terrains. As a theoretical novelty, TRAL formulates representation and apprenticeship learning in the unified regularized optimization framework to perform joint robot learning for terrain classification and behavior generation, which also automatically estimates the importance of terrain feature modalities. Our second theoretical novelty is the design of an optimization algorithm to solve the formulated problem, which possesses a theoretical convergence guarantee. The proposed approach is extensively evaluated using two real-world scenarios in the experiments, in which a mobile robot navigates over familiar or unfamiliar unstructured terrains. Experimental results have demonstrated that TRAL is able to transfer human expertise to mobile robots with small errors, obtain superior performance compared with previous and baseline methods, and offer intuitive insights on the importance of terrain feature modalities.

CHAPTER 5
NAUTS: NEGOTIATION FOR ADAPTATION TO UNSTRUCTURED TERRAIN SURFACES

This chapter is modified from a conference paper published at the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)[13]. This work was partially supported by NSF CAREER Award IIS-1942056 and U.S. DEVCOM ARL SARA Program W911NF-20-2-0107

Sriram Siva[14], Maggie Wigness[15], John Rogers[16], Long Quang[17], and Hao Zhang[18].

5.1 Abstract

When robots operate in real-world off-road environments with unstructured terrains, the ability to adapt their navigational policy is critical for effective and safe navigation. However, off-road terrains introduce several challenges to robot navigation, including dynamic obstacles and terrain uncertainty, leading to inefficient traversal or navigation failures. To address these challenges, we introduce a novel approach for adaptation by negotiation that enables a ground robot to adjust its navigational behaviors through a negotiation process. Our approach first learns prediction models for various navigational policies to function as a terrain-aware joint local controller and planner. Then, through a new negotiation process, our approach learns from various policies' interactions with the environment to agree on the optimal combination of policies in an online fashion to adapt robot navigation to unstructured off-road terrains on the fly. Additionally, we implement a new optimization algorithm that offers the optimal solution for robot negotiation in real-time during execution. Experimental results have validated that our method for adaptation by negotiation outperforms previous methods for robot navigation, especially over unseen and uncertain dynamic terrains.

[13] ©2022 IEEE/RSJ. Reprinted with permissions from Sriram Siva, Maggie Wigness, Long Quang, John G. Rogers, and Hao Zhang. "NAUTS: Negotiation for Adaptation to Unstructured Terrain Surfaces", in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS), 2022.

[14] Primary researcher and author, Graduate Student, Colorado School of Mines

[15] Senior Research Scientist, United States DEVCOM Army Research Lab

[16] Senior Research Scientist, United States DEVCOM Army Research Lab

[17] Researcher, United States DEVCOM Army Research Lab

[18] Associate Professor, Colorado School of Mines

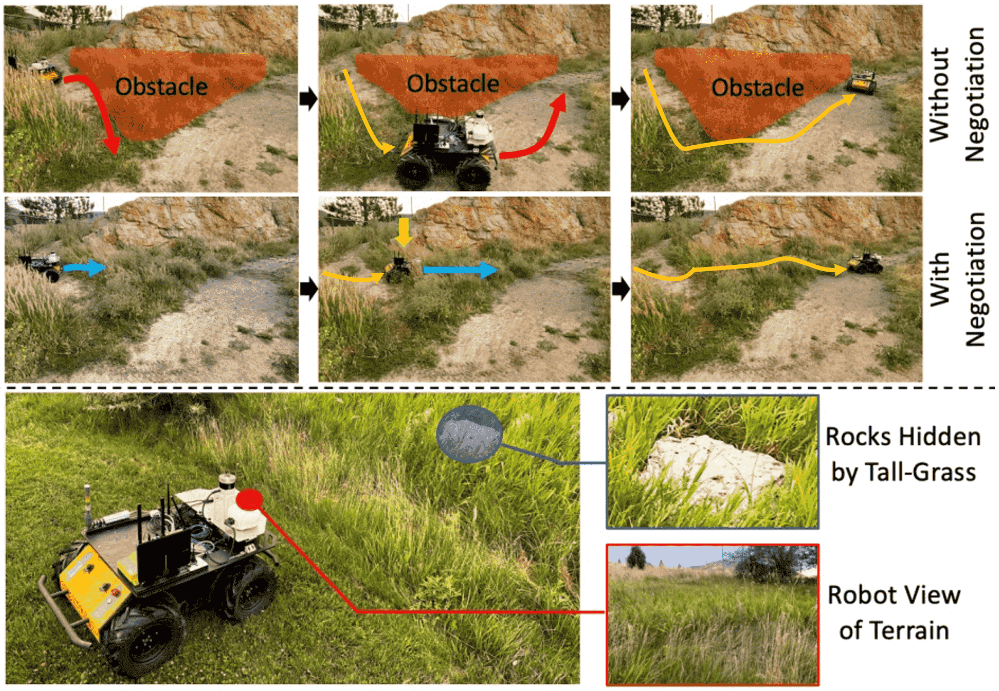


Figure 5.1 Robots operating in dynamic, unstructured environments often generate sub-optimal behaviors leading to inefficient robot traversal or even navigation failure. For example, in tall grass terrain, robots consider the terrain as an obstacle. Terrain negotiation allows robots to explore different navigation policies to determine the optimal combination for successful and efficient navigation in unknown terrains. In this example, the robot initially treats tall grass as an obstacle but simultaneously explores a max speed policy. The robot then quickly learns that the max speed policy improves efficiency by traversing across tall grass, learning to give more importance to max speed over obstacle avoidance.

5.2 Introduction

In recent years, autonomous mobile robots have been increasingly deployed in off-road field environments to carry out tasks related to disaster response, infrastructure inspection, and subterranean and planetary exploration [156–158]. When operating in such environments, mobile robots come across dynamic, unstructured terrains that offer a wide variety of challenges (as seen in Figure 5.1), including dynamic obstacles and varying terrain characteristics like slope and softness. In these environments, terrain adaptation is an essential capability that allows ground robots to perform successful maneuvers by adjusting their navigational behaviors to best traverse the changing unstructured off-road terrain characteristics [6, 159].

Given its importance, the problem of robot adaptation over unstructured terrains has been extensively investigated in recent years. In general, terrain adaptation has been addressed using three broad categories of methods. The first category, classic control-based methods, use mathematical tools from control theory [160–162] to design robot models that achieve the desired robot behavior and perform robust ground maneuvers in various environments. The second category, learning-based methods, use data-driven formulations to either imitate an expert demonstrator [6, 163, 164], learn from trial-and-error in a reinforcement learning setting [165–167], or use online learning to continuously learn and adapt in an environment [168–170]. Finally, the third category, machine-learning-based control, exploits the advantage of integrating machine learning into control theory to learn accurate robot dynamics and accordingly adapt navigational behaviors [171–173].

However, unstructured terrains often have dynamic obstacles that change their state as the robot traverses over them, such as tall grass. Additionally, these terrains can occlude future obstacles and ground cover, leading to traversal uncertainty (e.g., grass occluding a rock as seen in Figure 5.1). These challenges can also be observed in commonly traversed unstructured environments such as sand, snow, mud, and forest terrains. As characteristics of such terrains cannot be modeled beforehand, robots cannot be trained for all possible terrain variations and must therefore adapt as these variations are encountered. Existing methods for robot navigation generally lack robustness to address these challenges as they are designed as a local controller to execute a single robot navigation policy, causing inefficient (e.g., longer traversal time and distance) or even failed navigation. In addition, current methods such as [163, 164] require significant amount of training data to learn optimal navigational behaviors. The challenge of quickly learning a joint local controller and planner to enable adaptive behaviors has not been addressed.

In this chapter, we introduce our novel approach to robot navigation: *Negotiation for Adaptation to Unstructured Terrain Surfaces* (NAUTS). Instead of generating terrain-aware behaviors for only the current time steps, NAUTS learns a prediction model to estimate future robot behaviors and states for several different policies. Each policy represents a series of navigational behaviors that can be learned either using imitation learning [6] or self-supervised learning [164] according to a specific goal (e.g., obstacle avoidance, maximum speed, etc.). NAUTS then learns from the continuous interaction of these different policies with the terrain to generate optimal behaviors for successful and efficient navigation. We define *negotiation* as the process of learning robot navigation behaviors from online interactions between a library of policies with the terrain in order to agree on an optimal combination of these policies. The learning of both the prediction models and policy negotiation are integrated into a unified mathematical formulation under a regularized optimization paradigm.

There are three main contributions of this chapter:

- We introduce a novel learning-based prediction model to estimate goal-driven future robot behaviors and states according to various navigational policies, and address the challenge of learning a terrain-aware joint local controller and planner.
- We propose one of the first formulations on negotiation for robot adaptation under a regularized optimization framework. Our approach allows a robot to continuously form agreements between various navigational policies and optimally combines them to i) improve the efficiency of navigation in known environments and ii) learn navigation fast in unknown and uncertain environments.
- We design a new optimization algorithm that allows for fast, real-time convergence to execute robot negotiation during deployment.

As an experimental contribution, we provide a comprehensive performance evaluation of learning-based navigation methods over challenging dynamic unstructured terrains.

5.3 Related Work

The related research in robot terrain adaptation can be classified under methods based on classical control theory, learning-based, and machine-learning-based control.

The methods developed under the classical control theory use pre-defined models to generate robust navigational behaviors and reach the desired goal position in an outdoor field environment.

Earlier methods used a fuzzy logic implementation to perform navigation [174, 175], without using the knowledge of a robot's dynamics. This led to the development of system identification [176, 177], where methods learn robot dynamics using transfer functions to model linear robotic systems and perform navigation [178, 179]. More recently, trajectory optimization models such as differential dynamic programming (DDP), specifically iterative linear quadratic regulator (iLQR), used knowledge of non-linear robot dynamics to solve navigation tasks [180–182]. Model predictive control (MPC) learns to be robust to robot model errors and terrain noise by implementing a closed-loop feedback system during terrain navigation [183–185]. However, these methods can approximate robot dynamics to a limited extent as these methods cannot learn from high-dimensional robot data and lack the ability to adapt as terrain changes.

Learning-based methods use data-driven formulations to generate navigational behaviors in various environments. Early methods used Koopman operator theory [186] to model non-linear robot systems using an infinite-dimensional robot observable space [187, 188] to perform terrain navigation. Subsequent learning-based methods mainly used learning from demonstration (LfD) [189] to transfer human expertise of robot driving to mobile robots [117, 163, 190]. One method to perform terrain-aware navigation combined representation learning for terrain classification with apprenticeship learning to perform terrain adaptation [6]. Kahn and Levine [164] learned navigational affordance from experts over various terrains for carrying out off-road navigation. Recently, consistent behavior generation was achieved [34] to match actuation behaviors with a robot's expected behaviors. Unlike learning from demonstration, reinforcement learning based methods learn purely from a robot's own experience in an unknown environment in a trial-and-error fashion [165, 166, 191]. Rapid motor adaptation was achieved by updating learned policies via inferring key environmental parameters to successfully adapt in various terrains [167]. Life-long learning methods, similar to reinforcement learning, sequentially improve the performance of robot navigation by continuously optimizing learned models [170, 192]. Rather than just learning a robot model, learning-based methods also learn robot interactions with the terrain, thus being terrain-aware. However, these methods fail in unstructured environments [193] as they cannot adapt on the fly with the terrain or exhibit catastrophic forgetting [30], which is the tendency to forget previously learned data upon learning from new data.

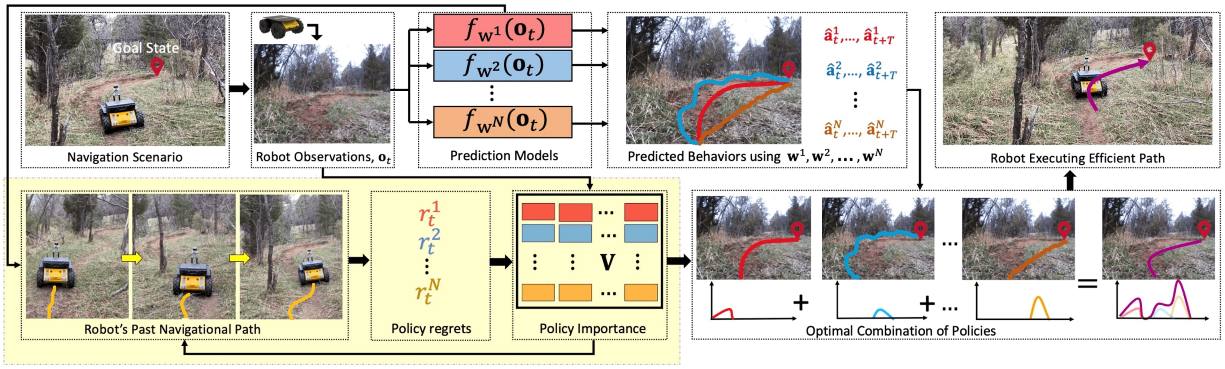

Goal State
Navigation Scenario
Robot Observations, $\mathbf{o}_t$
$f_{\mathbf{w}^1}(\mathbf{o}_t)$
$f_{\mathbf{w}^2}(\mathbf{o}_t)$
⋮
$f_{\mathbf{w}^N}(\mathbf{o}_t)$
Prediction Models
$\hat{\mathbf{a}}_t^1, \dots, \hat{\mathbf{a}}_{t+T}^1$
$\hat{\mathbf{a}}_t^2, \dots, \hat{\mathbf{a}}_{t+T}^2$
⋮
$\hat{\mathbf{a}}_t^N, \dots, \hat{\mathbf{a}}_{t+T}^N$
Predicted Behaviors using $\mathbf{w}^1, \mathbf{w}^2, \dots, \mathbf{w}^N$
Robot Executing Efficient Path
Robot's Past Navigational Path
r_t^1
r_t^2
⋮
r_t^N
Policy regrets
V
Policy Importance
+
+ ...
=
Optimal Combination of Policies

Figure 5.2 Overview of our proposed NAUTS approach for robot negotiation to adapt over unstructured terrains. Illustrated is the learning performed by our approach during the training phase. The module in the yellow box illustrates robot negotiation during the execution stage.

Machine-learning-based control methods learn robot behaviors by combining data-driven formulations into predefined robot models [194, 195]. Early methods used Dynamics Mode Decomposition (DMD) [196, 197] and Sparse Identification of Non-Linear Dynamics (SINDy) [198] to learn data-driven models based on system identification and performed terrain navigation [199, 200]. Later, evolutionary algorithms were developed to optimize parameters of a robot model in an online learning fashion for robust navigation [201, 202]. For robots with multiple degrees of freedom, methods were developed that use a combination of iterative Linear Quadratic Regulators (iLQR) and machine learning search to explore multiple robot configurations and plan self-adaptive navigation [203]. Similar approaches were designed using a neural network based functional approximator to learn a robot dynamics model and adapt this model with online learning [204]. Robust path planning was performed for safe navigation of autonomous vehicles under perception uncertainty [205]. However, these methods do not address adaptation to previously unseen, unstructured terrains, and are unable to address the dynamic nature of the terrain, which often leads to ineffective terrain traversal.

5.4 Approach

In this section, we discuss our proposed method, NAUTS, for robot traversal adaptation by negotiation. A overview of the approach is illustrated in Figure 5.2.

5.4.1 Learning Policy Prediction Models

Our approach first learns a prediction model to estimate future robot states and behaviors for each policy in a previously trained library. Navigational policies are the goal of the navigation, for example, obstacle avoidance, adaptive maneuvers or max speed navigation. This model enables our approach to predict how a policy works without the requirement of knowing its implementation (i.e., the policy can be treated as a black box). Formally, at time t, we denote the robot terrain observations (e.g., RGB images) input to the i-th policy as $\mathbf{o}_t^i \in \mathbb{R}^q$, where q is the dimensionality of the terrain observations. The robot behavior controls, i.e, navigational behaviors (e.g., linear and angular velocity), and states (e.g., robot's body pose and position) output from the policy are denoted as $\mathbf{a}_t^i \in \mathbb{R}^c$ and $\mathbf{s}_t^i \in \mathbb{R}^m$, with c and m denote the dimensionality of robot behaviors and states respectively. Then the i-th policy can be represented as $\pi^i : (\mathbf{s}_t^i, \mathbf{o}_t^i) \to \mathbf{a}_t^i$.

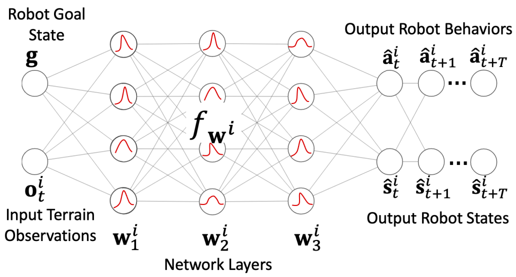

Robot Goal State
$\mathbf{g}$
$\mathbf{o}_t^i$
Input Terrain Observations
$\mathbf{w}_1^i$
$\mathbf{w}_2^i$
$\mathbf{w}_3^i$
Network Layers
$f_{\mathbf{w}^i}$
Output Robot Behaviors
$\hat{\mathbf{a}}_t^i$ $\hat{\mathbf{a}}_{t+1}^i$ ... $\hat{\mathbf{a}}_{t+T}^i$
$\hat{\mathbf{s}}_t^i$ $\hat{\mathbf{s}}_{t+1}^i$ ... $\hat{\mathbf{s}}_{t+T}^i$
Output Robot States

Figure 5.3 A shallow GP is designed to implement our prediction model $f_{\mathbf{w}^i}$.

Let g denote the relative goal state (with respect to $\mathbf{s}_t^i$) that the robot needs to reach at time $t+T$. For every policy π^i, we propose to learn a prediction model $f_{\mathbf{w}^i} : (\mathbf{o}_t^i, \mathbf{g}) \to (\hat{\mathbf{a}}_{t:t+T}^i, \hat{\mathbf{s}}_{t:t+T}^i)$ that predicts a sequence of goal driven T-future robot behaviors $\hat{\mathbf{a}}_{t:t+T}^i$ and states $\hat{\mathbf{s}}_{t:t+T}^i$. The prediction model estimates behaviors for the present time and functions like a local controller, and by estimating robot behaviors and states for future T-steps, it functions as a local planner. We introduce a shallow Gaussian Process (GP) [206] to implement $f_{\mathbf{w}^i}$ that is parameterized by $\mathbf{w}^i$, as shown in Figure 5.3. This shallow Gaussian Process with a recursive kernel has been shown in [206] is equivalent to, but more data-efficient than, a deep Bayesian CNN with infinitely many filters. In addition, as this Gaussian Process assumes that each weight of the network is a distribution instead of scalar values, it allows for uncertainty modeling and thus, is robust to environmental variations. We then learn the prediction model $f_{\mathbf{w}^i}$ by solving the following regularized optimization problem:

$$\min_{\mathbf{w}^i} \quad \lambda_1 \mathcal{L}\big((\pi^i(\mathbf{s}^i_{t:t+T}, \mathbf{o}^i_{t:t+T}), \mathbf{s}^i_{t:t+T}), f_{\mathbf{w}^i}(\mathbf{o}^i_t, \mathbf{g})\big) \quad + \lambda_2 \|\mathbf{g}^i - (\hat{\mathbf{s}}^i_{t+T} - \hat{\mathbf{s}}^i_t)\|_2^2 \tag{5.1}$$

where $\mathcal{L}(\cdot)$ is the cross-entropy loss [207], mathematically expressed as $\mathcal{L}(p, q) = -\mathbb{E}_p[\log(q)]$. This loss helps the prediction model to be insensitive to noisy observations in unstructured environments due to the logarithmic scale. The first part of Eq. (8.1) models the error of predicting T-future robot behaviors and states from actual navigational behaviors and states. The second part of Eq. (8.1) models the error of the robot failing to reach its relative goal state. The hyper-parameters λ_1 and λ_2 model the trade-off between the losses.

Following Eq. (8.1), the robot learns prediction models for N-different policies. However, when navigating over unstructured terrains, a single policy may not always prove to be effective for all scenarios. For example, the policy of obstacle avoidance may lead to longer traversal time in grass terrain, and the policy of max speed may cause collisions with occluded obstacles.

5.4.2 Robot Negotiation for Terrain Adaptation

The key novelty of NAUTS is its capability of negotiating between different policies to perform successful and efficient navigation, especially in unstructured off-road terrains. Given N-policies in the library, NAUTS formulates robot adaptation by negotiation under the mathematical framework of multi-arm bandit (MAB) optimization [208]. MAB comes from the hypothetical experiment where the robot must choose between multiple policies, each of which has an unknown regret with the goal of determining the best (or least regretted) outcome on the fly. We define regret, $r^i_t : (\mathbf{o}^i_{t-T}, \mathbf{g})\big) \to \mathbb{R}^+$, of the i-th policy at time t as the error of not reaching i) the goal position and ii) the goal position in minimum time and effort. We calculate the regret for each policy as:

$$r_t^i = \Big(\frac{\|\mathbf{g}\|_2\|\hat{\mathbf{s}}_t^i\|_2}{(\mathbf{g})^\top(\hat{\mathbf{s}}_t^i)} - 1\Big) + \sum_{k=t-T}^{t} (t-k)(\hat{\mathbf{a}}_k^i)^\top \hat{\mathbf{a}}_k^i \tag{5.2}$$

where the first part of Eq. (8.2) models the error of not reaching the goal position, with zero regret if the robot reached its goal position. This error grows exponentially if the robot has deviated from the goal position. The second part of Eq. (8.2) models the error of not reaching the goal in minimum time and effort. Specifically, the regret is smaller when the robot uses fewer values of navigational behaviors to reach the same goal and also if the robot reaches the goal in minimum time due to the scaling term $(t-k)$.

Unstructured terrain-aware negotiation can be achieved using the best subset of policies that minimize the overall regret in the present terrain as:

$$\begin{aligned} \min_{\mathbf{V}} \quad & \lambda_3 \sum_{i=1}^{N} \mathcal{R}(\mathbf{o}_t^i, r_t^i; \mathbf{v}^i) + \lambda_4 \|\mathbf{V}\|_E \\ \text{s.t.} \quad & \sum_{i=1}^{N} (\mathbf{o}_t^i)^\top \mathbf{v}^i = 1 \end{aligned} \tag{5.3}$$

where $\mathcal{R}(\cdot)$, parameterized by $\mathbf{v}^i \in \mathbb{R}^q$, is the terrain-aware regret of choosing policy π^i in the present terrain and $\mathbf{V} = [\mathbf{v}^1, \dots, \mathbf{v}^N] \in \mathbb{R}^{N\times q}$. Mathematically, $\mathcal{R}(\mathbf{o}_t^i, r_t^i; \mathbf{v}^i) = \sum_{k=t}^{t+T} \|r_k^* - (\mathbf{o}_t^i)^\top \mathbf{v}^i r_k^i\|_2^2$, with $r_k^* = \min r_k^i; i = 1, \dots, N$. The use of a linear model enables real-time convergence for terrain-aware policy negotiation. The column sum of $\mathbf{V}$ indicates the weights of each policy towards minimizing the overall regret of robot navigation. In doing so, the robot recognizes the important policies and exploits these policies for maintaining navigation efficiency. However, we also need to explore the various policies to improve navigation efficiency or even learn in an unknown environment, which is achieved by the regularization term in Eq. (5.3), called the exploration norm. Mathematically, $\|\mathbf{V}\|_E = \sum_{i=1}^{N} \frac{\|\mathbf{V}\|_F}{\|\mathbf{v}^i\|_2}$, where the operator $\|\cdot\|_F$ is the Frobenius norm with $\|\mathbf{V}\|_F = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{q}(v_j^i)^2}$. The exploration norm enables NAUTS to continuously explore all navigational policies in any terrain. Specifically, the exploration norm enables NAUTS to explore sub-optimal policies by ensuring $\mathbf{v}^i \neq \mathbf{0}$. If $\mathbf{v}^i = \mathbf{0}$, i.e., if the i-th policy is given zero importance, then the value of objective in Eq. (5.3) would be very high. The hyper-parameters λ_3 and λ_4 control the trade-off between exploration and exploitation during negotiation. The constraints in Eq. (5.3) normalize the various combination of navigational policies.

Integrating prediction model learning and policy negotiation under a unified mathematical framework, robot adaptation by negotiation can be formulated as the following regularized optimization problem:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{V}} \sum_{i=1}^{N} \Big(\lambda_1 \mathcal{L}\big((\pi^i(\mathbf{s}^i_{t:t+T}, \mathbf{o}^i_{t:t+T}), \mathbf{s}^i_{t:t+T}), f_{\mathbf{w}^i}(\mathbf{o}^i_t, \mathbf{g})\big) \\ &+ \lambda_2 \|\mathbf{g}^i - (\hat{\mathbf{s}}^i_{t+T} - \hat{\mathbf{s}}^i_t)\|_2^2 + \lambda_3 \mathcal{R}(\mathbf{o}^i_t, r^i_t; \mathbf{v}^i) \Big) + \lambda_4 \|\mathbf{V}\|_E \\ &\text{s.t.} \qquad \sum_{i=1}^{N} (\mathbf{o}^i_t)^\top \mathbf{v}^i = 1 \end{aligned} \tag{5.4}$$

where $\mathbf{W} = [\mathbf{w}^1, \dots, \mathbf{w}^N]$. During the training phase, we compute the optimal $\mathbf{W}^*$ and $\mathbf{V}^*$.

During execution, we fix $\mathbf{W}^*$, meaning the prediction models do not update during execution. However, our approach continuously updates $\mathbf{V}^*$ in an online fashion, which allows for negotiation at each step. At every time step t_0, we acquire observations $\mathbf{o}_{t_0}$. For a given robot goal state g, we dynamically choose the best combination of policies as:

$$\mathbf{a}_{t_0:t_0+T} = \sum_{i=1}^{N} (\mathbf{o}_{t_0})^\top \mathbf{v}^{i*} f_{\mathbf{w}^{i*}}(\mathbf{o}_{t_0}, \mathbf{g}) \tag{5.5}$$

where $\mathbf{a}_{t_0}$ is the behavior executed by the robot following policy negotiation at time t_0 and the behaviors $\mathbf{a}_{t_0:t_0+T}$ make up the local plan for the robot.

5.4.3 Optimization Algorithm

During training, we reduce Eq. (8.3) to simultaneously optimize $\mathbf{W}^*$ and $\mathbf{V}^*$. As the first term is non-linear, reducing Eq. (8.3) amounts to optimizing a non-linear objective function. We use the zeroth order non-convex stochastic optimizer from [209]. This optimizer has been proven to avoid saddle points and avoids local minima during optimization [209], and is specifically designed for

Algorithm 4: Optimization algorithm for solving the robot negotiation problem during execution in our objective function.

```
Input  : Policies $\mathbf{W}^*$ and Weights $\mathbf{V}^* \in \mathbb{R}^{N \times q}$
Output : Optimized Weights for Negotiation $\mathbf{V}^* \in \mathbb{R}^{N \times q}$
while goal is not reached do
    for $i = 1, \ldots, N$ do
        Obtain predicted behavior $\hat{\mathbf{a}}^i_{t:t+T}$ and states $\hat{\mathbf{s}}^i_{t:t+T}$ from $f_{\mathbf{w}^{i*}}(\mathbf{o}_{t_0}, \mathbf{g})$;
        Calculate regret of $i$-th policy $r^i$ from Eq. (8.2);
    Calculate $r^*_{t_0} = \min r^i_{t_0};\ i = 1, \ldots, N$;
    while not converge do
        Calculate diagonal matrix $\mathbf{Q}$ with the $i$-th diagonal block given as $\frac{\mathbf{I}}{2\|\mathbf{V}\|_E}$;
        Compute the columns of the distribution $\mathbf{V}$ according to Eq. (7.8);
return: $\mathbf{V}^* \in \mathbb{R}^{N \times q}$
```

constrained optimization problems like in Eq. (8.3). Additionally due to its weaker dependence on input data dimensionality [209], $\mathbf{W}$ and $\mathbf{V}$ can be computed faster despite using high dimensional terrain observations.

To perform robot adaptation by negotiation, we optimize $\mathbf{V}$ in an online fashion during the execution phase by solving the MAB optimization problem in Eq. (5.3), which has a convex objective with non-smooth regularization term. To perform fast online learning for negotiation, we introduce a novel iterative optimization algorithm that is tailored to solve the regularized optimization in Eq. (5.3), which at each time step performs fast iterations and converges in real-time to an optimal value of $\mathbf{V}$. The optimization algorithm follows the iterative re-weighted method [55] for optimizing the negotiation objective with non-smooth regularization term. This optimization algorithm is provided in Alg. 4. Specifically, to solve for the optimal $\mathbf{V}$, we minimize Eq. (5.3) with respect to $\mathbf{v}^i$, resulting in:

$$\sum_{k=t}^{t+T} \lambda_3 \big(2(r^i_k)^2 (\mathbf{o}^i_t)^\top (\mathbf{o}^i_t)\mathbf{v}^i - 2r^*_k r^i_k \mathbf{o}^i_t\big) + \lambda_4 \mathbf{Q}\mathbf{v}^i = 0 \tag{5.6}$$

where $\mathbf{Q}$ is a block diagonal matrix expressed as $\mathbf{Q} = \frac{\mathbf{I}}{2\|\mathbf{V}\|_E}$ and $\mathbf{I} \in \mathbb{R}^{N \times N}$ is an identity matrix. Then, we compute $\mathbf{v}^i$ in a closed-form solution as:

$$\mathbf{v}^i = (\lambda_4 \mathbf{Q} + 2\sum_{k=t}^{t+T} \lambda_3 (r^i_k)^2 (\mathbf{o}^i)^\top \mathbf{o}^i)^{-1} \lambda_3 \sum_{k=t}^{t+T} (2r^*_k r^i_k \mathbf{o}^i) \tag{5.7}$$

Because $\mathbf{Q}$ and $\mathbf{V}$ are interdependent, we are able to derive an iterative algorithm to compute them as described in Algorithm 4.

5.5 Proof of Convergence for the Optimization Algorithm

In the following, we prove that Algorithm 4 decreases the value of the objective function in Eq. (8.3) with each iteration during execution and converges to a local optimal solution.

At first, we present a lemma:

Lemma 4: For any two given vectors $\mathbf{a}$ and $\mathbf{b}$, the following inequality relation holds: $||\mathbf{b}||_2 - \frac{||\mathbf{b}||_2^2}{2||\mathbf{a}||_2} \leq ||\mathbf{a}||_2 - \frac{||\mathbf{a}||_2^2}{2||\mathbf{a}||_2}$

Proof.

$$-(||\mathbf{b}||_2 - ||\mathbf{a}||_2)^2 \leq 0$$

$$-||\mathbf{b}||_2^2 - ||\mathbf{a}||_2^2 + 2||\mathbf{b}||_2||\mathbf{a}||_2 \leq 0$$

$$2||\mathbf{b}||_2||\mathbf{a}||_2 - ||\mathbf{b}||_2^2 \leq ||\mathbf{a}||_2^2$$

$$||\mathbf{b}||_2 - \frac{||\mathbf{b}||_2^2}{2||\mathbf{a}||_2} \leq ||\mathbf{a}||_2 - \frac{||\mathbf{a}||_2^2}{2||\mathbf{a}||_2}$$

□

From Lemma 6, we can derive the following corollary:

Corollary 3: For any two given matrices $\mathbf{A}$ and $\mathbf{B}$, the following inequality relation holds:

$$||\mathbf{B}||_E - \frac{||\mathbf{B}||_E^2}{2||\mathbf{A}||_E} \leq ||\mathbf{A}||_E - \frac{||\mathbf{A}||_E^2}{2||\mathbf{A}||_E}$$

where the operator $||\cdot||_E$ is the exploration norm introduced in the main paper.

Theorem 4: Algorithm 5 converges fast to a local optimal solution to the terrain negotiation problem in Eq. (8.3) during execution.

Proof. According to Step 8 of Algorithm 5, for each iteration step s during optimization, the value of $\mathbf{v}^i(s+1)$ can be given as:

$$\mathbf{v}^i(s+1) = \|r^*(s+1) - (\mathbf{o}_t^i)^\top \mathbf{v}^{i*}(s+1) r^i(s+1)\|_2^2 \\ + \sum_{i=1}^{N} (\lambda_4 (\mathbf{v}^i(s+1))^\top \mathbf{Q}(s+1)(\mathbf{v}^i(s+1))) \tag{5.8}$$

where $\mathbf{Q}(s+1) = \frac{\mathbf{I}}{2\|\mathbf{V}(s)\|_E}$. Then we derive that:

$$\mathcal{J}(s+1) + \sum_{i=1}^{N} (\lambda_4 (\mathbf{v}^i(s+1))^\top \mathbf{Q}(s+1)(\mathbf{v}^i(s+1))) \\ \leq \mathcal{J}(s) + \sum_{i=1}^{N} (\lambda_4 (\mathbf{v}^i(s))^\top \mathbf{Q}(s)(\mathbf{v}^i(s))) \tag{5.9}$$

where $\mathcal{J}(s) = \|r^*(s) - (\mathbf{o}_t^i)^\top \mathbf{v}^{i*}(s) r^i(s)\|_2^2$.

After substituting the definition $\mathbf{Q}$ in Eq. (7.13), we obtain

$$\mathcal{J}(s+1) + (\lambda_4 \frac{\|\mathbf{V}(s+1)\|_E^2}{2\|\mathbf{V}(s)\|_E}) \leq \mathcal{J}(s) + (\lambda_4 \frac{\|\mathbf{V}(s)\|_E^2}{2\|\mathbf{V}(s)\|_E}) \tag{5.10}$$

From Corollary 4, for the weight matrix $\mathbf{V}$ we have:

$$\left(\|\mathbf{V}(s+1)\|_E - \frac{\|\mathbf{V}(s+1)\|_E^2}{2\|\mathbf{V}(s)\|_E} \right) \leq \left(\|\mathbf{V}(s)\|_E - \frac{\|\mathbf{V}(s)\|_E^2}{2\|\mathbf{V}(s)\|_E} \right). \tag{5.11}$$

Adding Eq. (7.14) and (7.15) on both sides, we have

$$\mathcal{J}(s+1) + \lambda_4 \leq \mathcal{J}(s) + \lambda_4 \|\mathbf{V}(s)\|_E \tag{5.12}$$

Eq. (7.16) implies that the updated value of weight matrix $\mathbf{V}$, decreases the value of the objective function with each iteration. As the negotiation problem in Eq. (8.3) is convex, Algorithm 1 4 converges to an optimal solution. Furthermore, during each time step of execution, we start with near-optimal $\mathbf{V}$ from previous time steps and as the objective is convex, Algorithm 4 converges faster than when starting from initial conditions, i.e., $\mathbf{V}$ as a zero matrix. □

Convergence: Algorithm 4 is guaranteed to converge to an optimal solution for the optimization problem in Eq. (5.3).

Complexity: For each iteration of Algorithm 4, computing Steps 3, 4, and 7 is trivial, and Step 8 is computed by solving a system of linear equations with quadratic complexity.

5.6 Experiments

This section presents the experimental setup and implementation details of our NAUTS approach, and provides a comparison of our approach with previous state-of-the-art methods.

5.6.1 Experimental Setup

We use a Clearpath Husky UGV robot for our field experiments. The robot is equipped with an Intel Realsense D435 color camera, an Ouster OS1-64 LiDAR, a Global Positioning System (GPS), and an array of sensors including a Microstrain 3DM-GX5-25 Inertial Measurement Unit (IMU) and wheel odometers. The robot states, i.e., robot pose, are estimated using an Extended Kalman Filter (EKF) [210], applied on sensory observations from LiDAR, IMU, GPS, and wheel odometers. The RGB images and the estimated robot states are used as our inputs. The robot runs a 4.3 GHz i7 CPU with 16GB RAM and Nvidia 1660Ti GPU with 6GB of VRAM, which runs the behavior prediction models at 5Hz and policy negotiation at 0.25 Hz.

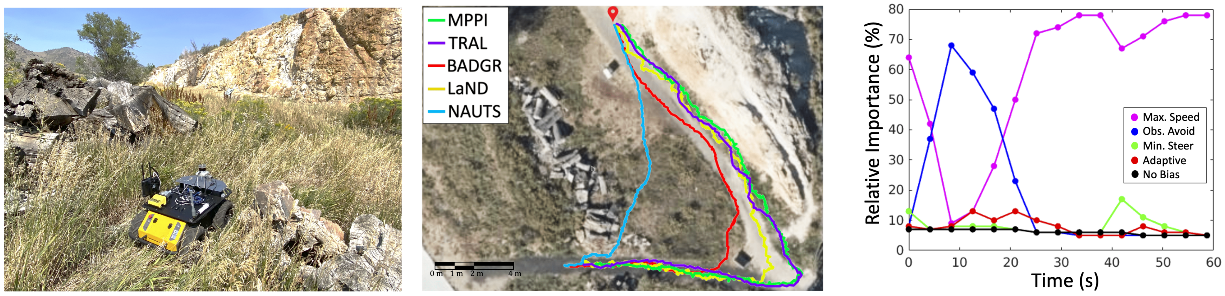
We evaluate our approach on navigation tasks that require traversing from the robot's initial position to a goal position, and provide a performance comparison against state-of-the-art robot navigation techniques including Model Predictive Path Integral (MPPI) [161] control, Terrain Representation and Apprenticeship Learning (TRAL) [6], Berkley Autonomous Driving Ground Robot (BADGR) [164], and Learning to Navigate from Disengagements (LaND) [163]. To quantitatively evaluate and compare these approaches to NAUTS, we use the following evaluation metrics:

- *Failure Rate (FR)*: This metric is defined as the number of times the robot fails to complete the navigation task across a set of experimental trials. If a robot flips or is stopped by a terrain obstacle, it is considered a failure. Lower values of FR indicate better performance.

- *Traversal Time (TT)*: This metric is defined as the time taken to complete the navigation task over given terrain. Smaller values of TT indicate better performance.

- *Distance traveled (DT)*: This metric is defined as the total distance traveled by the robot when completing a navigational task. A smaller DT value may indicate better performance.

- *Adaptation time (AT)*: This metric is defined as the time taken by the robot to regain half its linear velocity when introduced to an unseen unstructured environment. A lower value of AT may indicate better performance.



(a) Tall grass terrain (b) Paths taken by different methods (c) Importance of different policies

Figure 5.4 The tall grass terrain used in our experiments and the qualitative results over this terrain.

To collect the training data, a human expert demonstrates robot driving over simple terrains of concrete, short grass, gravel, medium-sized rocks, large-sized rocks and forest terrain. Each of these terrain were used to learn one specific aspect of robot navigation such as adjusting traversal speeds over large-sized rocks, or obstacle avoidance using the forest terrain. Specifically, we used these terrains to learn from a library of five distinct navigational policies:

- *Maximum Speed:* When following this navigational policy, the human expert drives with the maximum traversal speed irrespective of the terrain the robot traverses upon. The aim when following the maximum speed navigational policy is to teach the robot to cover as much distance as possible in the least amount of time. Thus, while collecting training data with this policy the expert demonstrator uses straight line traversal without steering the robot.

- *Obstacle Avoidance:* While following this policy, the expert demonstrates how to maneuver by driving around obstacles to avoid collision. To learn this policy, expert demonstrations in forest terrains are used where humans navigate the forest by avoiding trees and logs while moving the robot through the terrain. The underlying goal with this policy is to teach the robot to steer around obstacles.

- *Minimum Steering:* For this policy, the expert drives the robot with limited steering. During navigation, linear velocity is fixed to 0.75 m/s and obstacle avoidance is performed by beginning to turn the robot when it is further away from obstacles instead of making short, acute turns. The policy differs from obstacle avoidance by maintaining a fixed speed while taking a smooth and long maneuver around obstacles.

- *Adaptive Maneuvers:* While following this policy, the expert varies the robot's speed across different terrain to reduce traversal bumpiness. Specifically, with terrains that are relatively less rugged such as concrete or short-grass, the expert demonstrator uses high speed maneuvers. On the other hand, over terrains with high ruggedness such as gravel or medium sized rocks, the expert demonstrator uses slower speeds, with the slowest traversal speed across the large rocks terrain.

- *No Navigational Bias:* When following this policy, multiple expert demonstrators navigate the robot across the different terrains without particular policy bias, i.e., without following any specific navigational policy. The underlying goal behind using such policy is to cover most of the common navigational scenarios encountered by the robot, and include the navigational bias from multiple expert demonstrators.

For each policy, the robot is driven on each of the different terrains, resulting in approximately 108000 distinctive terrain observations with the corresponding sequence of robot navigational behaviors and states for each navigational policy. No further pre-processing is performed on the collected data. We use this data to learn optimal π^i, $i = 1, \ldots, N$ and $\mathbf{V}$ during training. We learn these parameters for different values of hyper-parameters of the NAUTS approach, i.e., λ_1, λ_2, λ_3, λ_4 and T. The combination of these hyper-parameters that results in the best performance of

NAUTS during validation are then used for our experiments. In our case, the optimal performance of NAUTS is obtained at $\lambda_1 = 0.1, \lambda_2 = 10$, $\lambda_3 = 1$ and $\lambda_4 = 0.1$ for $T = 9$.

Table 5.1 Quantitative results for scenarios when the robot traverses over dynamic, uncertain grass terrain.

Metrics	MPPI	TRAL	BADGR	LaND	**NAUTS**
Failure Rate (/10)	3	3	**1**	5	**1**
Traversal Time (s)	88.72	72.99	64.47	90.18	**58.79**
Distance Traveled (m)	68.58	56.69	50.29	64.93	**36.57**
Adaptation Time (s)	14.23	10.92	-	-	**6.24**

5.6.2 Navigating over Dynamic Uncertain Grass Terrain

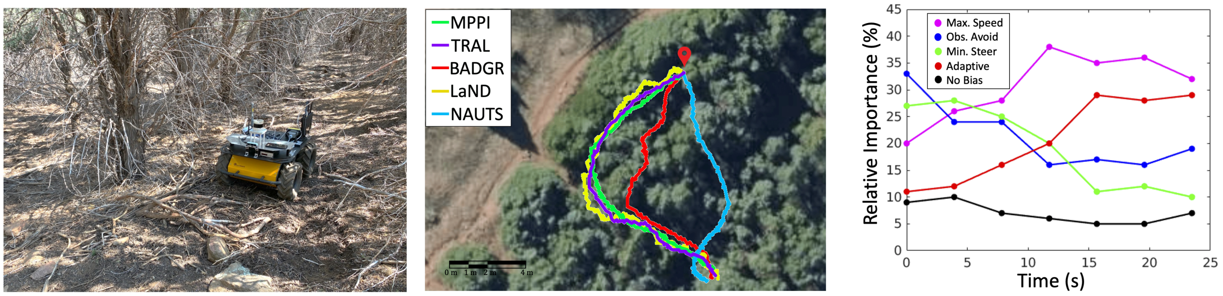
In this set of experiments, we evaluate robot traversal performance over the tall grass terrain environment, as shown in Figure 5.4(a). This is one of the most commonly found terrains in off-road environments and is characterized by deformable dynamic obstacles added with the terrain uncertainty of occluded obstacles. The process of negotiation is continuously performed throughout the execution phase. The evaluation metrics for each of the methods are computed across ten trial runs over the tall grass terrain environment.

The quantitative results obtained by our approach and its comparison with other methods are presented in Table 5.1. In terms of the FR metric, BADGR and NAUTS obtain the lowest values, whereas MPPI, TRAL and LaND have high FR values. Navigation failure for MPPI, TRAL and LaND generally occurred as the robot transitioned into the tall grass terrain where it would get stuck after determining the tall grass was an obstacle. Failure cases for NAUTS and BADGR occurred when the robot was stuck in the tall grass terrain due to wheel slip. Both NAUTS and BADGR obtain significantly fewer failures than MPPI and LaND methods due to their ability to adapt to different terrains.

When comparing the traversal time and the distance traversed by the different methods, we observe that NAUTS obtains the best performance followed by BADGR and TRAL. The LaND and MPPI approaches obtain higher TT and DT metrics, with MPPI performing the poorest in terms of DT and LaND performing poorest in terms of TT. A qualitative comparison, from a single trial, of the path traversed by these methods is provided in Figure 5.4(b). Notice, MPPI, LaND, and TRAL all consider tall grass as obstacles and avoid this terrain while traversing. We observe that BADGR and NAUTS explore tall grass terrain and the shortest path is taken with our NAUTS approach resulting in the lowest DT and TT values.

NAUTS also performs better than the TRAL and MPPI approaches in terms of the AT metric. The AT metric is observed when robots encounter an unseen terrain and require adaptation. In this environment, that happens when the robot transitions into the tall grass terrain. We do not provide AT values for BADGR and LaND as both approaches have a fixed linear velocity without adaptation. Overall, we observe that our approach obtains successful navigation (from FR metric) and better efficiency (from TT and DT metrics) over previous methods.

Figure 5.4(c) illustrates the NAUTS negotiation process between the five policies in the tall grass terrain. NAUTS learns optimal combinations of policies in real-time during execution (each update is marked by dots in the figure). Initially, max speed has higher importance over other policies. However, as the robot enters tall grass, obstacle avoidance becomes more important. While traversing further, the robot learns to give more importance to the max speed policy again and obstacle avoidance becomes less important. All other policies have relatively low importance, but they never reach zero, as NAUTS regularly evaluates the different policies.



(a) Forest terrain (b) Path taken by different methods (c) Importance of different policies

Figure 5.5 The forest terrain used in our experiments and the qualitative results over this terrain.

5.6.3 Navigating on Unseen Unstructured Forest Terrain

In this set of experiments, we evaluate navigation across forest terrains. Apart from high uncertainty and dynamic obstacles, this terrain has different characteristics that the robot has not previously seen during training, e.g, terrain covered with wood chips, dried leaves, rocks, and tree branches. Similar to the previous set of experiments, the evaluation metrics in the forest terrain are computed across ten runs for each of the methods.

The quantitative results over off-road forest terrain are presented in Table 7.2. In terms of the FR metric, we observe a similar trend seen in the tall grass terrain experiments. Specifically, MPPI and TRAL have similar performance in terms of FR metrics. Our NAUTS approach obtains the lowest FR value followed by the BADGR approach, and the LaND approach obtains the highest value. Common failures in the forest terrain occur when tree branches occluding the terrain are classified as obstacles or traversing over large rocks, wooden tree barks, or mud in the terrain cause the robot to get stuck. NAUTS also obtains better efficiency in both the TT and DT metrics, followed by the BADGR approach. Again, MPPI and TRAL both obtain similar TT and DT values, and LaND exhibits the worst performance.

Figure 5.5(b) illustrates qualitatively how MPPI, TRAL, and LaND avoid uncertain and unseen paths and follow an existing trail. However, BADGR explores unknown paths, reaching the goal faster than these methods, and NAUTS outperforms all methods by exploring different policies in this unseen terrain. In this set of experiments, the AT metric is observed throughout navigation as each section of the terrain is not previously seen by the robot and requires the robot to adapt. NAUTS obtains better AT values than MPPI and TRAL, indicating a better adaptation capability.

Table 5.2 Quantitative results for scenarios when the robot traverses over unseen dynamic, unstructured off-road forest terrain.

Metrics	MPPI	TRAL	BADGR	LaND	**NAUTS**
Failure Rate (/10)	5	5	4	7	**2**
Traversal Time (s)	34.28	33.95	26.17	33.98	**24.21**
Distance Traveled (m)	24.68	23.77	20.94	26.51	**16.45**
Adaptation Time (s)	10.04	11.93	-	-	**7.32**

Figure 5.5(c) illustrates the negotiation process by NAUTS in the forest terrain. At the start of the navigation task, each policy has different importance, with obstacle avoidance being the most significant. As the robot continues with the navigation task, it learns to use the optimal combination

of policies, which results in the most efficient navigation. Thus, the max speed and adaptive navigational policies become more significant than other policies. It is important to note that there is no single optimal policy throughout navigation due to i) the highly unstructured nature of this terrain and ii) the continuous exploration of the NAUTS approach.

5.7 Summary

In this chapter, we introduce the novel NAUTS approach for robot adaptation by negotiation for navigating in unstructured terrains, that enables ground robots to adapt their navigation policies using a negotiation process. Our approach learns a prediction model to function as a terrain-aware joint local controller and planner corresponding to various policies, and then uses the negotiation process to form agreements between these policies in order to improve robot navigation efficiency. Moreover, our approach explores different policies to improve navigation efficiency in a given environment continuously. We also developed a novel optimization algorithm that solves an optimal solution to the robot negotiation problem in real-time. Experimental results have shown that our approach enables a robot to negotiate its behaviors with the terrain and delivers more successful and efficient navigation compared to previous methods.

CHAPTER 6

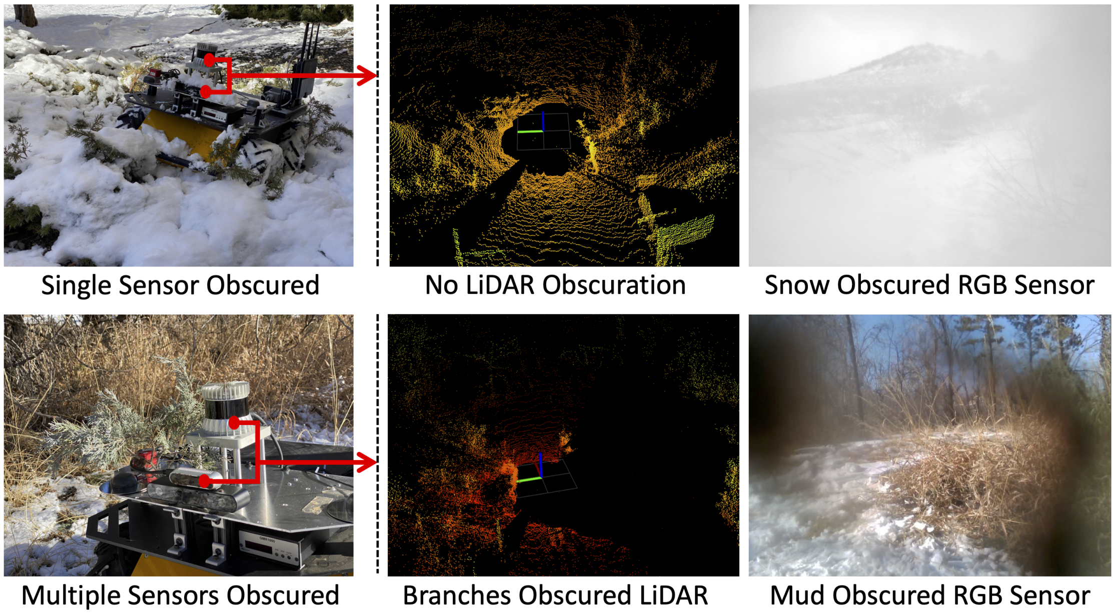
CAPABILITY-AWARE PERCEPTUAL ADAPTATION FOR ROBUST GROUND NAVIGATION IN UNSTRUCTURED OFF-ROAD ENVIRONMENTS

This work was partially supported by NSF CAREER Award IIS-1942056 and the DEVCOM ARL SARA Program W911NF-20-2-0107.

Perceptual adaptation is crucial for autonomous ground robots as it enables them to adjust to perceptual changes in real-world environments to successfully complete their fundamental functionalities like navigation. However, when operating in these unstructured environments robots often encounter perceptual challenges such as obscuration or even failure in one or more of its sensors, which negatively affect the robot's traversability and cause inaccurate or even failed navigation. To address this shortcoming, we propose a novel human-inspired method of perceptual adaptation. Our approach uses multi-sensory observations to develop capability awareness in robot perception in a self-supervised fashion and also learns the robot's dynamics model to generate robust navigational behaviors. Specifically, our approach learns modality-invariant representations, i.e., a common trajectory representation, from multiple sensor modalities, which allows the robot to be aware of perceptual obscurations and failures, and accordingly adapt its navigational behaviors. Furthermore, we also implement a new optimization algorithm to solve the formulated optimization problem of learning capability-aware perceptual adaptation. To evaluate our approach, we perform extensive experimentation using a physical ground robot with real-world scenarios of sensor obscurations and simulated sensor failures over various unstructured off-road terrains. Experimental results prove that our method can better adapt to these perceptual challenges when navigating over unstructured terrains and outperform previous and baseline methods.

6.1 Introduction

Autonomous mobile robots have been increasingly used over the past several decades in real-world applications to traverse over complex unstructured off-road terrains, including disaster response, search and rescue, and subterranean and planetary exploration [158, 211–213]. Such field environments are challenging for ground robots due to their unstructured nature. In addition, these environments present a wide variety of perceptual challenges, including sensor obscuration, low lighting, lens flares, etc., which cannot be modeled beforehand, as can be seen from scenarios in Figure 6.1. The ability of ground robots to adjust their behaviors with such perceptual challenges, i.e., perceptual adaptation [10, 214], is essential to their successful traversal in unstructured field environments.



Single Sensor Obscured

No LiDAR Obscuration

Snow Obscured RGB Sensor

Multiple Sensors Obscured

Branches Obscured LiDAR

Mud Obscured RGB Sensor

Figure 6.1 Motivational images to develop capability awareness in robot navigation. As seen from the figure, robots navigating in unstructured environments often encounter perceptual challenges like sensor obscurations or failures. Such challenges may cause errors in robot localization and inaccuracy or even failed navigation. Therefore, the capability of ground robots to be capability-aware and adapt to these perceptual challenges is essential to the successful completion of the navigation task.

Given the importance of robot navigation, numerous terrain adaptation and perceptual adaptation methods have been proposed in recent years. Terrain adaptation methods use either a) tools from control theory to develop robot models from the robot's internal states [161, 162, 215, 216], b) use data-driven formulations, i.e. learning-based, to perform terrain navigation a robot's perception [6, 34, 165, 192, 217] or c) use machine-learning-based control to exploit the advantages of integrating robot learning into control theory and generate robust navigational behaviors [171–173, 203]. Perceptual adaptation methods allow a robot to continuously adjust its navigational behaviors based on perceptual inputs from various sensors in unstructured environments [10, 218–220]. More recently, robots that develop awareness have been proposed, that continuously monitor perceptual inputs from various sensors to be conscious of their environment and execute navigational behaviors [221–223].

However, previous approaches lack the ability to adapt to the perceptual challenges of sensor obscuration or failures. Examples of sensor obscurations include robot sensors partially covered by mud or snow, lens flares, or occlusions due to tree branches, whereas failures in perception include robot sensors completely covered by mud, snow, or even broken sensors. As can be seen from Figure 6.1, such scenarios are commonly encountered when ground robots operate in real-world field environments and cannot always be predicted or anticipated. Therefore, robots cannot be trained for all possible scenarios and need to adapt to these perceptual challenges. Existing learning-based methods generally ignore these challenges. Specifically, most of the existing methods lack the ability to be aware of sensory obscurations or failures, and thus cannot adapt to such scenarios which often leads to inaccurate or even failed robot navigation. The challenge of generating robust navigational behaviors under perceptual challenges for successful navigation in unstructured environments has not been well addressed.

To address this challenge, we develop a novel human-inspired method of learning capability awareness that learns to be conscious of perceptual challenges and adapts to these challenges to execute robust navigational behaviors. We use the term *capability awareness* to refer to robots' ability to recognize and predict their limitations (e.g., failed or obscured sensor), and accordingly, adapt their behaviors while being aware of the result of these behaviors. Our approach draws inspiration from the fact that humans can estimate their movements (with limited accuracy) even when their eyes are closed, using other sensing modalities in the body such as the vestibular system for detecting movement [224] and skin receptors for detecting touch and pressure [225]. To achieve capability awareness in perception, we learn a modality-invariant trajectory representation that encodes the robot's trajectory information in a smaller observation space. We use the term trajectory representation to define the latent states, but not necessarily a physical quantity, achieved by the robot when traversing over a terrain. Modality-invariant representations allow our approach to estimate a common trajectory representation from all the sensory modalities and still encode important terrain information. In order to adapt to perceptual challenges, our approach continuously monitors the trajectory representations from different sensory modalities to identify the obscured/failed sensors and accordingly estimates trajectory representations from functional sensors. Our approach then uses trajectory representations to learn the robot's dynamics model which estimates the robot's future trajectory based on its present navigational behaviors. Learning of both the capability awareness and perception-aware dynamics model are integrated into a unified mathematical formulation under the mathematical framework of self-supervised learning.

The key novelty of our approach is the introduction of a human-inspired method to learn capability awareness in perception, which advances the state-of-the-art by enabling a ground robot to be resilient to sensor obscurations or failures and accordingly execute navigational behaviors to complete a navigational task over unstructured terrains. The specific novelty of our approach includes:

- We propose a novel approach to learning capability awareness in robot perception that under the self-supervised learning framework, which enables a robot to be conscious of perceptual obscuration or failures through learning a modality-invariant trajectory representation.

- We introduce a novel method of modeling the dynamics of the robot, that uses modality-invariant trajectory representations to estimate how the robot's trajectory changes according to the executed navigational behaviors.

- We also propose a new optimization algorithm to effectively converge to an optimal solution despite the dependent variable and non-smooth regularization terms present in the formulated optimization problem.

The remainder of the paper is organized as follows. a review of the related work is provided in Section 6.2. our proposed approach is introduced in Section 6.3. We provide the experimental results and comparisons in Section 6.5 and finally conclude the paper in Section 6.6.

6.2 Related Work

This section reviews the related research on robot terrain navigation and perceptual adaptation.

6.2.1 Robot Terrain Navigation

Methods developed under the arch of robot terrain navigation can be broadly classified under the category of classical control, learning-based, and machine learning-based control.

Methods developed under the classical control theory use pre-defined models to generate robust navigational behaviors and reach the desired goal position in an outdoor field environment. Many of the earlier methods used a fuzzy logic-based implementation [226, 227], without prior knowledge of the robot's dynamics to perform terrain navigation. Then system identification [228] models were developed that used robot trajectory data to learn robot dynamics and accordingly perform terrain navigation [229, 230]. Subsequently, trajectory optimization for terrain navigation was achieved either through differential dynamics programming (DDP) [231, 232] or iterative linear quadratic regulators (iLQR) [233, 234], both of which learn non-linear robot dynamics model from robot's interaction with terrain. Closed loop feedback control was achieved via Model Predictive Control (MPC) [235, 236] and Model Predictive Path Integral (MPPI) [161] control that is robust to terrain noise and error in robot models as well.

Learning-based methods use data-driven formulations to generate navigational behaviors in various environments. Earlier methods modeled non-linear robot systems using high-dimensional linear observation space [187, 188]. Subsequent learning-based methods mainly used learning from demonstration (LfD) [80, 237] to transfer expert demonstrations to a ground robot to traverse over various terrains [117, 192]. Navigational affordances were learned from experts over various terrains in [164] for carrying out off-road navigation. Recently, consistent behavior generation was achieved by [34] to match actuation behaviors with the robots' expected behaviors. Unlike learning from demonstration, reinforcement learning-based methods purely learn from the robot's own experience in an unknown environment in a hit-and-trial fashion [165–167, 191]. Life-long learning methods, similar to reinforcement learning, sequentially improve the performance of robot navigation by continuously optimizing learned models [170, 192, 238].

Machine learning-based control methods learn to generate navigational behaviors both by combining data-driven formulations into predefined robot models [194, 195, 239, 240]. Earlier methods used Sparse Identification of Non-Linear Dynamics (SINDy) [198, 241] to learn data-driven models based on system identification and performed terrain navigation [199, 200]. Then evolutionary algorithms were developed to optimize parameters of a robot model in an online learning fashion for robust navigation [201, 202]. For robots with multiple degrees of freedom, methods were developed that use a combination of iterative Linear Quadratic Regulators (iLQR) and machine learning search to explore multiple robot configurations and plan self-adaptive navigation [203, 242]. Although the above methods were successfully learned to navigate in outdoor environments, they rely on perceptual observations from various sensors and fail during obscuration or failure, thus lacking the ability to adapt on the fly.

6.2.2 Robot Perceptual Adaptation

The methods developed under robot perceptual adaptation can be classified under methods of learning from perception and methods that learn awareness.

The methods that learn from perception aim reliably operate in an unstructured environment in the presence of perceptual challenges. Earlier methods used perception for learning sub-tasks within the navigation, for example, image segmentation [243, 244] and object detection [245, 246]. To address the problem of unstructured terrain navigation [247] classified terrains of different types and accordingly adjusted the speed through terrain classification. Then methods were used to generate digital terrain maps and then fitted a Gaussian model to classify the terrains [155, 248] and adjust their traversing velocities with these terrains. A few techniques were also designed to use terrain ruggedness data to evaluate navigational behaviors in an online fashion for ground vehicles

[111, 249]. In learning-based methods, terrain classification was usually performed by learning from terrain data to classify the terrain for navigational behaviors. Support Vector Machine (SVM) based classifier was used on features learned from Hidden Markov Models to identify terrains [132]. Color-based terrain classification was performed to label obstacles and generate navigational behaviors [133]. More recently, semantic segmentation methods based on deep learning [250] were successfully used to classify off-road terrains [251, 252] and perform the task of terrain navigation.

The methods that learn awareness generate an understanding of the robot such that it can recognize and predict its capabilities, and accordingly adjust its navigational behaviors in an environment. Robust path planning was performed in [205] for the safe navigation of autonomous vehicles by perceiving the various uncertainties in its perception. Multi-sensory observations were used to detect vegetation-obscured obstacles during navigation in [220, 253]. Perceptual awareness [254, 255] based on 2D-LiDAR sensors were used in [256] to learn navigation in cluttered environments. Awareness through environmental lighting conditions was used by [10], to calibrate perception in dynamically changing environments and perform navigation. Recently, situational awareness models were used by [257, 258] to adapt navigational behaviors in robots based on the robot's perception.

However, they still lack the ability to adapt to sensor failures and more importantly fail to execute behaviors for successful navigation during sensor failure. Furthermore, even methods that use multi-sensory information for robust navigation during sensor failure [259], significantly drop their performance under perceptual failure or obscurations[259]. In this paper, we propose to maintain the performance of the robot under perceptual failures.

6.3 Approach

In this section, we discuss our novel human-inspired method for capability-aware perceptual adaptation to enable robust ground navigation in unstructured off-road environments. An overview of our approach is shown in Figure 6.2.

Notations: We denote vectors with boldface lowercase letters, e.g., $\mathbf{m} \in \mathbb{R}^p$. We write matrices using boldface capital letters, e.g., $\mathbf{M} = \{m_i^j\} \in \mathbb{R}^{p \times q}$, and we refer to its i-th row and j-th column as $\mathbf{m}^j$ and $\mathbf{m}_i$. We use boldface capital Euler script letters to denote a set of matrices, e.g., $\boldsymbol{\mathcal{M}} = \{\mathrm{M}_1, \ldots, \mathrm{M}_r\}$, and $|\cdot|$ to denote its cardinality, e.g., $|\boldsymbol{\mathcal{M}}| = r$.

6.3.1 Learning Capability Awareness in Perception

Our approach learns capability awareness through the learning of modality-invariant trajectory representations. These representations are used to encode the robot trajectory information from each of the sensor modalities in a much smaller observation space. We assume that multiple sensors are installed on a ground robot to collect both exteroceptive observations (such as RGBD images and LiDAR readings) and proprioceptive observations (such as wheel odometry and Inertial Measurement Unit (IMU) measurements) as the robot navigates over the unstructured terrain. We represent observations from the i-th sensor modality as $\mathbf{o}_i^{(t)} \in \mathbb{R}^{p_i}$ at time t, and p_i denotes the dimensionality of the modality. We represent the i-th modality's observations from a sequence of past c time steps as $\mathbf{O}_i^{(t)} = [\mathbf{o}_i^{(t)}, \ldots, \mathbf{o}_i^{(t-c)}] \in \mathbb{R}^{p_i \times c}$, and further, denote the multi-sensory observations from m-different sensory modalities as $\boldsymbol{\mathcal{O}}^{(t)} = \{\mathrm{O}_1^{(t)}, \ldots, \mathrm{O}_m^{(t)}\}$, where $|\boldsymbol{\mathcal{O}}| = m$ denotes the number of sensor modalities. We denote the trajectory representations from each of the i-th sensor modalities as $\mathrm{s}_i^{(t)} \in \mathbb{R}^r$, with r as its dimensionality of the trajectory representations. Then, $\mathrm{s}^{(t)} \in \mathbb{R}^r$ denotes the common trajectory representation from different sensors.

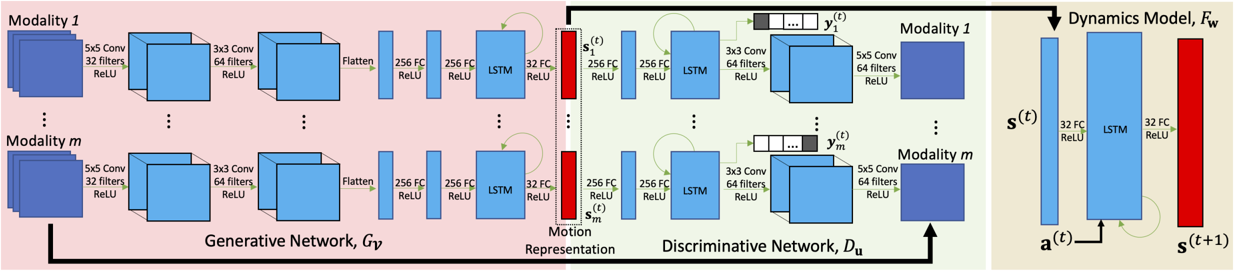


Figure 6.2 Overview of our proposed approach. From the left, the first module of the figure illustrates the generative network used to represent the generator model, the second module illustrates the discriminative network used to represent the discriminator network and the third module shows the shallow network used to represent the dynamics model of the robot.

Formally, given multi-sensory observations $\boldsymbol{\mathcal{O}}^{(t)}$, we propose to learn a generative model $G : \boldsymbol{\mathcal{O}}^{(t)} \rightarrow \mathbf{s}^{(t)}$ that generates modality-invariant trajectory representations $\mathbf{s}^{(t)}$, using $\boldsymbol{\mathcal{O}}^{(t)}$. To implement the generative model G, we use a set of m-different deep generative networks $G_{\mathbf{v}_i} : \mathbf{O}_i^{(t)} \rightarrow \hat{\mathbf{s}}_i^{(t)}$, each parameterized by weights $\mathbf{v}_i$, and are used to estimate trajectory representation $\hat{\mathbf{s}}_i^{(t)}$, using the i-th sensor modality $\mathbf{O}_i^{(t)}$. We collectively denote the m-different weights of generative networks as $\boldsymbol{\mathcal{V}} = \{\mathbf{v}_1, \dots, \mathbf{v}_m\}$. The model of our generative networks is shown in Figure 6.2, where the dimension of the input layer is varied based on the dimension of the individual sensory observations. The input sensory observations are processed using convolutional layers, followed by fully connected layers. The output of this layer is then processed using a recurrent neural network, which is essential to model the robot's continuous trajectory from a sequence of past c-observations.

We propose to learn capability awareness in perception through the learning of a modality invariant representation, which is achieved by simultaneously learning a discriminator model opposite to our generative model. Collectively, the aim of the m-different generative networks is to ensure that trajectory representations from each sensor modality are identical, i.e., $\hat{\mathbf{s}}_i^{(t)} = \hat{\mathbf{s}}_j^{(t)}; i \neq j$, whereas, the discriminator model aims to differentiate these trajectory representations to each of the sensor modalities and learns capability awareness in perception. Specifically, we learn capability awareness through learning modality invariant representations, when the generative model aims to fool the discriminator model by generating modality invariant trajectory representations to an extent where the discriminator model cannot differentiate between the representations.

The discriminator model $D : G_{\boldsymbol{\mathcal{V}}}(\boldsymbol{\mathcal{O}}^{(t)}) \rightarrow \{\mathbf{y}_{1:m}^{(t)}, \mathbf{o}_{1:m}^{(t)}\}$ processes the output of the generative model and generates m-different one-hot encoding vector $\mathbf{y}_i^{(t)} \in \mathbb{R}^m$ and m-different robot sensory observations $\mathbf{o}_{1:m}^{(t)}$. The one-hot encoding vector $\mathbf{y}_i^{(t)}$ is used to discriminate the trajectory representations from different sensor modalities and is used to indicate the robot's capability awareness in perception. Specifically, each vector, $\mathbf{y}_i^{(t)}$ has exactly one entry equal to one and all others equal to zero. The vector with i-th element as one indicates if the trajectory representation was generated by the i-th sensor modality. Then our approach generates capability awareness such that in case of sensory obscuration/failure, the discriminator model recognizes which obscured/failed sensor modality generated a representation as it can discriminate this representation to the modality which generated this representation. Apart from estimating $\mathbf{y}_{1:m}^{(t)}$, D also retrieves the sensory observations for m-different sensor modalities $\mathbf{o}_{1:m}^{(t)}$, from i-th trajectory representation $\mathbf{s}_i^{(t)}$. This allows our approach to retrieve sensory observations in case of a sensory failure while also ensuring the representations generated by G are informative enough of the terrain.

To implement the discriminator model D, we use a deep discriminative network $D_{\mathbf{u}} : G_{\boldsymbol{\mathcal{V}}}(\boldsymbol{\mathcal{O}}^{(t)}) \to \{\hat{\mathbf{y}}_{1:m}^{(t)}, \hat{\mathbf{o}}_{1:m}^{(t)}\}$, parameterized by weights $\mathbf{u}$, that monitors each of the m-different trajectory representation and discriminates them by accurately classifying which sensor modality generated the i-th trajectory representation $\hat{\mathbf{s}}_i^{(t)}$, i.e., to $\mathbf{y}_i^{(t)}$. The choice of our model for the discriminator network is illustrated in Figure 6.2, with a) fully connected input layers to process the trajectory representation from the generative network, b) intermediate recurrent layers that uses subsequent trajectory representation to accurately discriminate which faulty sensory modalities and finally c) convolutional layer capable of generating high-dimensional sensory observations, e.g., color images, with output layers mapped according to the dimension of the sensor observation being retrieved.

Then the problem of learning capability awareness in perception via learning of modality-invariant trajectory representations can be given as:

$$\min_{\boldsymbol{\mathcal{V}},\mathbf{u}} \quad \mathcal{L}_G\big(\boldsymbol{\mathcal{O}}^{(t)}, \mathbf{y}_{1:m}^{(t)}\big) + \mathcal{L}_D\big(\boldsymbol{\mathcal{O}}^{(t)}, \mathbf{y}_{1:m}^{(t)}\big) \tag{6.1}$$

where $\mathcal{L}_G$ and $\mathcal{L}_D$ represent the loss of the generative networks $G_{\boldsymbol{\mathcal{V}}}$ and the discriminative network $D_{\mathbf{u}}$, respectively. The generative loss is mathematically expressed as $\mathcal{L}_G\big(\boldsymbol{\mathcal{O}}^{(t)}, \mathbf{y}_i^{(t)}\big) = \sum_{i=1}^m \big(\lambda_1 \|\hat{\mathbf{y}}_i^{(t)} - (\mathbf{1}_m - \mathbf{y}_i^{(t)})\|_2^2\big)$, where $\mathbf{1}_m \in \mathbb{R}^m$ is a vector of ones that encodes the error of generating modality invariant representations. The vector $(\mathbf{1}_m - \mathbf{y}_i^{(t)})$ has all elements as one except the i-th element as zero and represents a scenario where the discriminative network $D_{\mathbf{u}}$, identifies $\hat{\mathbf{s}}_i^{(t)}$ to be generated from all the other sensor modalities except the i-th sensor modality $\mathbf{O}_i^{(t)}$. In Eq. (8.1), the discriminative loss $\mathcal{L}_D\big(\boldsymbol{\mathcal{O}}^{(t)}, \mathbf{y}_i^{(t)}\big) = \sum_{i=1}^m \big(\lambda_1 \|\hat{\mathbf{y}}_i^{(t)} - \mathbf{y}_i^{(t)}\|_2^2 + \lambda_2 \|\hat{\mathbf{o}}_i^{(t)} - \mathbf{o}_i^{(t)}\|_2^2\big)$ has two different objectives. The first part of the loss learns capability awareness in perception by exactly discriminating which sensor modality generated respective trajectory representations. Specifically, the error increases if the discriminator incorrectly discriminates the trajectory representations. The second part of the loss encodes the error of retrieving individual sensor observations, i.e., $\hat{\mathbf{o}}_i^{(t)}$ from $\hat{\mathbf{s}}_i^{(t)}$. Here, λ_1 and λ_2, denote the trade off-hyperparameters to balance between i) learning of capability awareness through learning modality-invariant trajectory representations and ii) learning to retrieve sensory observations.

Following Eq. (8.1), the robot can be aware of sensory obscurations or failures. However, it still cannot be used to complete a navigational task despite perceptual obscuration or failures, which is required for successful navigation in the wild.

6.3.2 Learning Robot Dynamics Model

We introduce in our approach a novel formulation of learning the robot's dynamics based on robot perception, to predict how the trajectory of the robot changes based on its executed navigational behaviors. Formally, we denote the robot's navigational behaviors, i.e., behavior controls (e.g., linear and angular velocity) as $\mathbf{a}^{(t)} \in \mathbb{R}^p$, with p as the dimensionality of the robot's navigational behaviors. During training, these navigational behaviors are recorded by an expert demonstrator only to learn the dynamics model. Then, given modality-invariant trajectory representation $\mathbf{s}^{(t)}$, we learn robot's dynamics model as $F : (\mathbf{s}^{(t)}, \mathbf{a}^{(t)}) \to \mathbf{s}^{(t+1)}$ to estimate the trajectory representation $\mathbf{s}^{(t+1)}$, at the next time step $t+1$, based on the present representation $\mathbf{s}^{(t)}$ and the present navigational behaviors $\mathbf{a}^{(t)}$.

To implement the dynamics model F, we use a shallow neural network $F_{\mathbf{w}} : (\hat{\mathbf{s}}_i^{(t)}, \mathbf{a}^{(t)}) \to \hat{\mathbf{s}}_i^{(t+1)}$, parameterized by weights $\mathbf{w}$ to predict the future trajectory representations $\hat{\mathbf{s}}_i^{(t+1)}$ from i-th sensory modality. The present trajectory representations for the i-th modality $\hat{\mathbf{s}}_i^{(t)}$, are obtained from the generative networks as $\hat{\mathbf{s}}_i^{(t)} = G_{\mathbf{v}_i}(\mathbf{O}_i^{(t)})$. Our choice of shallow network is illustrated in the third module of Figure 6.2, which consists of a fully connected input layer to process the present trajectory representation and a a recurrent network to model the past sequence of representations along with the navigational behaviors to output the future trajectory representations, thus allowing us to model the non-linear dynamics of the robot using a small order trajectory representation of the robot.

Then the problem of learning a robot's dynamics model from modality invariant trajectory representations can be given as:

$$\min_{\mathbf{w}} \mathcal{L}_F\Big(G_{\boldsymbol{\mathcal{V}}}(\mathbf{O}_{1:m}^{(t+1)}), F_{\mathbf{w}}(\hat{\mathbf{s}}_{1:m}^{(t)}, \mathbf{a}^{(t)})\Big) \tag{6.2}$$

where $\mathcal{L}_F\Big(G_{\boldsymbol{\mathcal{V}}}(\mathbf{O}_{1:m}^{(t+1)}), F_{\mathbf{w}}(\hat{\mathbf{s}}_{1:m}^{(t)}, \mathbf{a}^{(t)})\Big) = \sum_{i=1}^{m}\Big(\lambda_3 \|G_{\mathbf{v}_i}(\mathbf{O}_i^{(t+1)}) - F_{\mathbf{w}}(\hat{\mathbf{s}}_i^{(t)}, \mathbf{a}^{(t)})\|_2^2\Big)$ denotes the loss of the dynamics model and encodes the error of accurately modeling the robot dynamics to the trajectory representations of the generator model. In Eq. (8.2), λ_3 is a hyper-parameter to control the accuracy of learned dynamics model.

Integrating learning of modality invariant trajectory representation and learning of dynamics model into a unified formulation, the problem of learning capability-aware perceptual adaptation for robust navigation can be formulated as:

$$\begin{aligned} \min_{\boldsymbol{\mathcal{V}},\mathbf{u},\mathbf{w}} \quad & \mathcal{L}_G\big(\boldsymbol{\mathcal{O}}^{(t)}, \mathbf{y}_{1:m}^{(t)}\big) + \mathcal{L}_D\big(\boldsymbol{\mathcal{O}}^{(t)}, \mathbf{y}_{1:m}^{(t)}\big) \\ & + \mathcal{L}_F\Big(G_{\boldsymbol{\mathcal{V}}}(\mathbf{O}_{1:m}^{(t+1)}), F_{\mathbf{w}}(\hat{\mathbf{s}}_{1:m}^{(t)}, \mathbf{a}^{(t)})\Big) + \mathcal{R}_G(\boldsymbol{\mathcal{V}}) \end{aligned} \tag{6.3}$$

where, $\mathcal{R}_G(\boldsymbol{\mathcal{V}}) = \sum_{i=1}^{m} \frac{\lambda_4}{\sum_{j=1}^{p} a_j^{(t)} \|G_{\mathbf{v}_i}(\mathbf{O}_i^{(t+1)}) - G_{\mathbf{v}_i}(\mathbf{O}_i^{(t)})\|_2^2}$ is a regularization term on the generative model to ensure the successive trajectory representations generated from the generative networks are discriminate, with λ_4 denoting the hyper-parameter to control the amount of regularization. The regularization term ensures that the generative model is not generating a constant representation at different times and ensures that the successive representations are distinctive. This regularization term conditioned on the robot's navigational behaviors adds a high cost to our objective function if the robot is navigating and the successive trajectory representations are similar.

6.3.3 Robot Executing Perceptual Adaptation during Navigation

During the training phase, we compute the optimal weights $\boldsymbol{\mathcal{V}}^*$, $\mathbf{u}^*$ and $\mathbf{w}^*$ according to Algorithm 7. At each time step t, given the multi-sensory robot observation set $\boldsymbol{\mathcal{O}}^{(t)}$, the robot estimates modality-invariant trajectory representations from the generative networks as $\mathbf{s}_{1:m}^{(t)} = G_{\boldsymbol{\mathcal{V}}^*}(\boldsymbol{\mathcal{O}}^{(t)})$ and capability awareness using $\mathbf{y}_{1:m}^{(t)} = D_{\mathbf{u}^*}(G_{\boldsymbol{\mathcal{V}}^*}(\boldsymbol{\mathcal{O}}^{(t)}))$. For any of the i-th modality, if $\{\max_j(\mathbf{y}_i^{j(t)})\} \geq \gamma$, we identify these sensory as potentially obscured or failed, where γ is a manually defined threshold. We use trajectory representation $\mathbf{s}^{(t)*} = \mathbf{s}_i^{(t)}$ to perform robot navigation via the dynamics model, where $\mathbf{s}_i^{(t)}$ corresponds to awareness $\mathbf{y}_i^{(t)}$ with $\min(\max_j(\mathbf{y}_{1:m}^{j(t)}))$.

Once the robot obtains the optimal trajectory representation $\mathbf{s}^{(t)*}$, it estimates the trajectory representations for next step using the dynamics model as $\mathbf{s}^{(t+1)} = F_{\mathbf{w}^*}(\mathbf{s}^{(t)*}, \mathbf{a}^{(t)})$. Let $\mathbf{g} \in \mathbb{R}^q$ denote the robot's goal, i.e., desired pose and position relative to its present position, given to robot to reach at time $t + T$, with q as the dimensionality of the goal position. Then for a given goal $\mathbf{g}$, we use the dynamics model to generate optimal navigational behaviors by solving the dynamic programming problem similar to [257] as:

$$\begin{aligned} & \mathbf{a}^{*(t:t+N-1)} = \mathcal{L}_A(\mathbf{s}^{(k)}, \mathbf{a}^{(k)}, \mathbf{g}; \mathbf{s}^{(t)*}) \\ \text{s.t.} \qquad & \mathbf{a}_{min} \leq \mathbf{a}^{(k)} \leq \mathbf{a}_{max} \end{aligned} \tag{6.4}$$

where $\mathcal{L}_A(\mathbf{s}^{(k)}, \mathbf{a}^{(k)}, \mathbf{g}; \mathbf{s}^{(t)*}) = \|\mathbf{g} - \mathbf{P}(\mathbf{s}^{(t+N)} - \mathbf{s}^{*(t)})\|_2^2 + \sum_{k=t}^{t+N-1}(k-t)(\mathbf{a}^{(k)})^\top \mathbf{a}^{(k)}$; denotes the loss for estimating a sequence of optimal navigational behaviors $\mathbf{a}^{*(t:t+N-1)}$, to reach $\mathbf{g}$. The matrix $\mathbf{P} \in \mathbb{R}^{q \times r}$ is a fixed projection matrix, from robot's state space to its trajectory representation space and is estimated similar to [257]. The first part of the loss encodes the error of reaching its goal position in the $t + N$-th time and the second part of the loss ensures that the robot reaches the goal in minimum time due to the scaling factor of $(k - t)$. The constraints in Eq. (8.4) represent the limits on the robot's navigational behaviors, with $\mathbf{a}_{min}$ and $\mathbf{a}_{max}$ as its minimum and maximum navigational behaviors. We use a linear quadratic regulator (LQR) [260] based trajectory optimization to generate the optimal navigational behaviors.

Algorithm 5: Optimization algorithm for solving the capability-aware navigation problem in

```
Input  : Observation sets $\{\boldsymbol{\mathcal{O}}^{(t_1)}, \dots, \boldsymbol{\mathcal{O}}^{(t_n)}\}$ and Navigational behaviors $\{\mathbf{a}^{(t_1)}, \dots, \mathbf{a}^{(t_n)}\}$
Output : Optimized Weights for Generator Network $\boldsymbol{\mathcal{V}}^*$, Discriminator Network $\mathbf{u}^*$, and Dynamics
         Network $\mathbf{w}^*$
Initialize $\boldsymbol{\mathcal{V}}^*$, $\mathbf{u}^*$ and $\mathbf{w}^*$
while not converge do
    for $i = 1, \dots, m$ do
        Calculate the gradient of the objective in Eq. (8.3) w.r.t $\mathbf{v}_i$, i.e., $\nabla_{\mathbf{v}_i(s)}\mathcal{J}$ from Eq. (8.7);
        Update the weights of the generative network as: $\mathbf{v}_i(s+1) = \mathbf{v}_i(s) - \alpha_1(\nabla_{\mathbf{v}_i(s)}\mathcal{J})$;
    Calculate the gradient of the objective in Eq. (8.3) w.r.t $\mathbf{u}$, i.e., $\nabla_{\mathbf{u}(s)}\mathcal{J}$ from Eq. (8.11);
    Update the weights of the generative network as: $\mathbf{u}(s+1) = \mathbf{u}(s) - \alpha_2(\nabla_{\mathbf{u}(s)}\mathcal{J})$;
    Calculate the gradient of the objective in Eq. (8.3) w.r.t $\mathbf{w}$, i.e., $\nabla_{\mathbf{w}(s)}\mathcal{J}$ from Eq. (8.11);
    Update the weights of the generative network as: $\mathbf{w}(s+1) = \mathbf{w}(s) - \alpha_3(\nabla_{\mathbf{w}(s)}\mathcal{J})$;
return: $\boldsymbol{\mathcal{V}}^*$, $\mathbf{u}^*$ and $\mathbf{w}^*$
```

During navigation, if the robot encounters any obscuration or failure in perception, it obtains trajectory representation from the remaining sensor modalities and is updated at each time step. Our approach can obtain trajectory representation even with multiple obscurations and/or failed sensory modalities. Thus, even with these perceptual challenges the robot continues to successfully navigate towards its goal position.

6.4 Optimization Algorithm

The optimization problem in Eq. (8.3) is challenging to solve in general due to the dependent variables of: i) generative model, and ii) discriminative and dynamics model. In addition, the optimization problem also has a non-smooth regularization term, thus cannot be differentiated at non-smooth points during optimization, second-order optimization algorithms, such as Newton's method [261] are not applicable. To address the non-smooth regularization terms and dependent variables, we design a new alternating minimization algorithm that simultaneously learns the generative, discriminative and the dynamics model.

Formally, during the training phase, we iterative optimize between the i) generative model and ii) discriminative model and dynamics model. Specifically, during the iterative process, we first optimize the generative model, i.e, first and fourth terms in Eq. (8.3). Then, we fix the parameters for the generative model and optimize the discriminative and the dynamics model, i.e., the second and the third term in Eq. (8.3). To optimize the generative model, we calculate the gradient of our objective in Eq. (8.3) with respect to weights $\boldsymbol{\mathcal{V}}$ as:

$$\begin{aligned}\nabla_{\boldsymbol{\mathcal{V}}}\mathcal{J} =& \frac{d}{d\boldsymbol{\mathcal{V}}}\Bigg(\mathcal{L}_G\big(\boldsymbol{\mathcal{O}}^{(t)}, \mathbf{y}_{1:m}^{(t)}\big) + \mathcal{L}_D\big(\boldsymbol{\mathcal{O}}^{(t)}, \mathbf{y}_{1:m}^{(t)}\big) \\ & + \mathcal{L}_F\Big(G_{\boldsymbol{\mathcal{V}}}(\mathbf{O}_{1:m}^{(t+1)}), F_{\mathbf{w}}(\hat{\mathbf{s}}_{1:m}^{(t)}, \mathbf{a}^{(t)})\Big) + \mathcal{R}_G(\boldsymbol{\mathcal{V}})\Bigg)\end{aligned} \tag{6.5}$$

where $\mathcal{J}$ is used to represent the objective in Eq. 8.3. Since only the first and last term in Eq. (8.3) are optimized while learning the generator model, Eq. (8.5) can be optimized as:

$$\nabla_{\mathbf{v}_i}\mathcal{J} = \frac{d}{d\mathbf{v}_i}\Bigg(\mathcal{L}_G\big(\boldsymbol{\mathcal{O}}^{(thet)}, \mathbf{y}_{1:m}^{(t)}\big) + +\mathcal{R}_G(\boldsymbol{\mathcal{V}})\Bigg) \tag{6.6}$$

Substituting the expressions for the generator loss and regularization in Eq. 8.5, we obtain the gradient as:

$$\begin{aligned}\nabla_{\mathbf{v}_i}\mathcal{J} =& \lambda_1 \frac{d}{d\mathbf{v}_i}\Bigg(\big(D_{\mathbf{u}}(G_{\mathbf{v}_i}(\mathbf{O}^{(t)}))\big)^\top D_{\mathbf{u}}(G_{\mathbf{v}_i}(\mathbf{O}^{(t)}))\big) \\ & - \Big(\big(D_{\mathbf{u}}(G_{\mathbf{v}_i}(\mathbf{O}^{(t)}))\big)^\top (\mathbf{1}_m - \mathbf{y}_i^{(t)})\Big) \\ & - \Big((\mathbf{1}_m - \mathbf{y}_i^{(t)})^\top D_{\mathbf{u}}(G_{\mathbf{v}_i}(\mathbf{O}^{(t)}))\big)\Big)\Bigg) \\ & + \lambda_4 \frac{d}{d\mathbf{v}_i}\left(\frac{1}{\sum_{j=1}^{p} a_j^{(t)} \|G_{\mathbf{v}_i}(\mathbf{O}_i^{(t)}) - \hat{\mathbf{s}}_i^{(t-1)}\|_2^2}\right)\end{aligned} \tag{6.7}$$

At each iteration step s, we update the weights $\mathbf{v}_i$ for each of the generative network as:

$$\mathbf{v}_i(s+1) = \mathbf{v}_i(s) - \alpha_1(\nabla_{\mathbf{v}_i(s)}\mathcal{J}) \tag{6.8}$$

where α_1 denotes the learning rate for the generator network.

Once the weights of the generator network are updated, we then learn the discriminator model and the dynamics model, we calculate the gradient of our objective with respect to weights $\mathbf{u}$ and $\mathbf{w}$

as:

$$\nabla_{\mathbf{u}}\mathcal{J} = \frac{d}{d\mathbf{u}}\Big(\mathcal{L}_D\big(\boldsymbol{\mathcal{O}}^{(t)}, \mathbf{y}^{(t)}_{1:m}\big)\Big) \tag{6.9}$$

and

$$\nabla_{\mathbf{w}}\mathcal{J} = \frac{d}{d\mathbf{w}}\Big(\mathcal{L}_F\Big(G_{\boldsymbol{\mathcal{V}}}(\mathbf{O}^{(t+1)}_{1:m}), F_{\mathbf{w}}(\hat{\mathbf{s}}^{(t)}_{1:m}, \mathbf{a}^{(t)})\Big)\Big) \tag{6.10}$$

Substituting the expressions for the discriminator loss and dynamics loss in Eq. 8.10 and Eq. 6.10, we obtain the gradient as:

$$\begin{aligned}\nabla_{\mathbf{u}}\mathcal{J} =& \lambda_1 \frac{d}{d\mathbf{u}}\Big(\big(D_{\mathbf{u}}(G_{\mathbf{v}_i}(\mathbf{O}^{(t)}))\big)^\top D_{\mathbf{u}}(G_{\mathbf{v}_i}(\mathbf{O}^{(t)})) \\ &- \big(D_{\mathbf{u}}(G_{\mathbf{v}_i}(\mathbf{O}^{(t)}))\big)^\top \mathbf{y}_i^{(t)} - (\mathbf{y}_i^{(t)})^\top D_{\mathbf{u}}(G_{\mathbf{v}_i}(\mathbf{O}^{(t)}))\Big) \\ &+ \lambda_2 \frac{d}{d\mathbf{u}}\Big(\big(D_{\mathbf{u}}(G_{\mathbf{v}_i}(\mathbf{O}^{(t)}))\big)^\top D_{\mathbf{u}}(G_{\mathbf{v}_i}(\mathbf{O}^{(t)})) \\ &- \big(D_{\mathbf{u}}(G_{\mathbf{v}_i}(\mathbf{O}^{(t)}))\big)^\top \mathbf{o}_i^{(t)} - (\mathbf{o}_i^{(t)})^\top D_{\mathbf{u}}(G_{\mathbf{v}_i}(\mathbf{O}^{(t)}))\Big)\end{aligned} \tag{6.11}$$

and

$$\begin{aligned}\nabla_{\mathbf{w}}\mathcal{J} =& \lambda_3 \frac{d}{d\mathbf{w}}\Big(\big(F_{\mathbf{w}}(\hat{\mathbf{s}}_i^{(t)}, \mathbf{a}^{(t)})\big)^\top \big(F_{\mathbf{w}}(\hat{\mathbf{s}}_i^{(t)}, \mathbf{a}^{(t)})\big) \\ &- \big(F_{\mathbf{w}}(\hat{\mathbf{s}}_i^{(t)}, \mathbf{a}^{(t)})\big)^\top \hat{\mathbf{s}}_i^{(t+1)} - (\hat{\mathbf{s}}_i^{(t+1)})^\top F_{\mathbf{w}}(\hat{\mathbf{s}}_i^{(t)}, \mathbf{a}^{(t)})\Big)\end{aligned} \tag{6.12}$$

Accordingly, at each iteration step s, we update the weights of the discriminator and dynamics model as:

$$\mathbf{u}(s+1) = \mathbf{u}(s) - \alpha_2(\nabla_{\mathbf{u}(s)}\mathcal{J}) \tag{6.13}$$

and

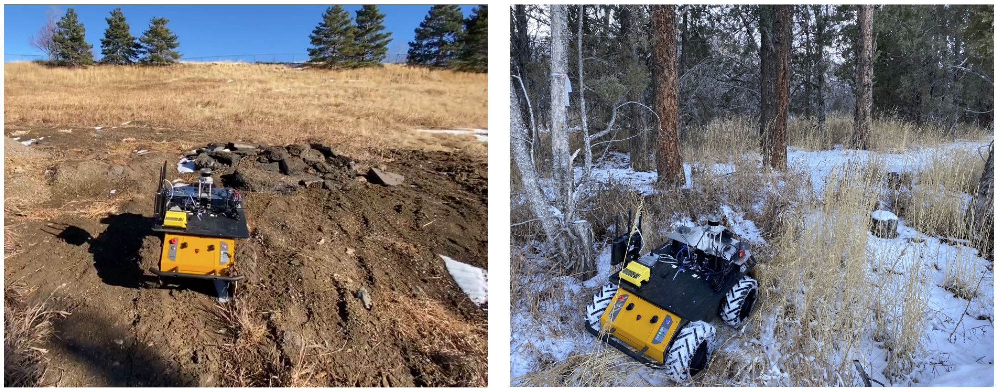
$$\mathbf{w}(s+1) = \mathbf{w}(s) - \alpha_3(\nabla_{\mathbf{w}(s)}\mathcal{J}) \tag{6.14}$$

where α_2 and α_3 are the learning rates for the discriminator model and the dynamics model respectively.

The discriminator model updated from Eq. (8.13) is used to then learn the generator model in Eq. (8.9). Then, we develop an iterative algorithm to solve the formulated optimization problem, which is described in Algorithm 7.

6.5 Experiments

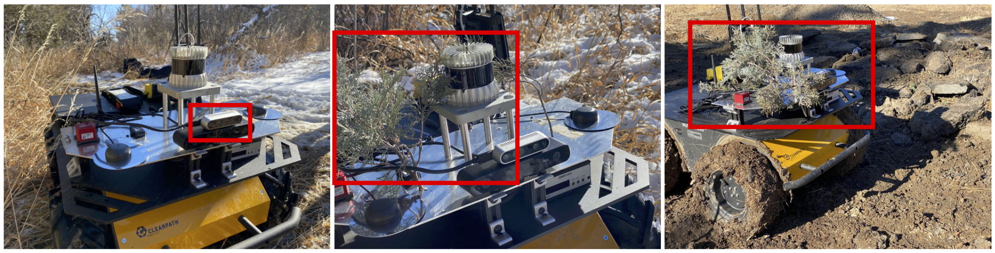
To assess the performance of our approach, we perform extensive experiments over real-world unstructured terrains. In this section, we discuss the experimental setups, and implementation, and finally, we present and analyze the experimental results of our approach in comparison to the other state-of-the-art methods.



Mixed Terrain

Forest Terrain

Figure 6.3 Unstructured terrains in the wild used for our experimentation.



Obscured RGB Sensor

Obscured LiDAR Sensor

Obscured RGB and LiDAR Sensor

Figure 6.4 The three different scenarios of sensor obscurations used in our experiments.

6.5.1 Experimental Setups

We use a Clearpath Husky ground robot to perform robot navigation over unstructured terrains. The robot is equipped with an Intel Realsense D435 color camera, a Stereo Labs Zed2i stereo RGB-D camera, an Ouster OS1-64 LiDAR, SBG Systems Ellipse-D GNSS with GPS and Inertial Measurement Unit (IMU), and wheel odometers. The a) RGB images from the color camera, b) the depth image from the stereo camera, c) elevation maps [262] from LiDAR data to represent robot centric grid-wise elevation of the terrain, and d) robot pose and position estimated using Extended Kalman Filtering (EKF) [210], applied on sensory observations from LiDAR, IMU, GPS, and wheel odometers are used as our inputs. Linear interpolation is used to obtain the inputs at a steady rate of 15Hz. The robot's onboard computer runs a 4.3 GHz Intel i7 processor, 32 GB memory with Nvidia GTX 1650 Ti GPU with 6GB VRAM, and runs our approach at 5Hz.

We evaluate our approach on the navigation task of traversing from the robot's initial position to a goal position in a real-world scenario. The two different scenarios include i) an unstructured mixed terrain, and ii) a forest terrain, as seen in Figure 6.3. Both of these terrains are commonly found environments when robots operate in the wild and present their respective challenges. Specifically, the unstructured mixed terrain consists of multiple terrain types including tall grass, sand, gravel, and large rocks with varying slopes and roughness. When navigating over such terrains, robots often encounter sensor obscuration due to tall grass or mud-covered sensors, or sensor failure, which make it challenging to navigate over this unstructured terrain. The forest terrain consists of varying terrain types of snow, tall grass, and snow-covered tree branches and presents challenges of multiple sensor obscuration from tall grass, tree branches, and environment clutter, as can be seen in Figure 6.3. For navigation tasks in each of the terrains, the goal position is set approximately ten meters away from the robot's initial position.

We provide a comparison of our approach with the state-of-the-art robot navigation techniques including the Terrain Representation and Apprenticeship Learning (TRAL) [6], Enhancing Consistent Ground Maneuverability (ECGM) [34], and Berkeley Autonomous Driving Ground Robot (BADGR) [164]. To quantitatively evaluate the performance of our approach we use the following metrics:

- *Failure Rate (FR)*: This metric is defined as the number of times the robot fails to complete the navigation task across a set of experimental trials. If a robot flips, gets stuck in a terrain, or is unable to navigate to the goal position, it is considered a failure. Lower values of FR indicate better performance and are preferred. Failures are manually observed and labeled.
- *Traversal Time (TT)*: This metric is defined as the average time taken to complete the navigational task in all the successful runs over the given terrain. Generally, smaller values of traversal time indicate better performance and may be preferred.

During training, we observe the best performance of our approach when using a sequence of $c = 5$ and hyperparameters $\lambda_1 = 0.1$, $\lambda_2 = 0.01$ $\lambda_3 = 0.1$ and $\lambda_4 = 1$. The threshold for perceptual awareness is set as $\gamma = 0.3$. The optimal number of future time steps used for executing the navigational task is set as $N = 9$.

Table 6.1 Quantitative results for scenarios when the robot encounters sensor obscurations. Successful runs (with no navigation failures) are used to calculate the metrics of traversal time and adaptation time.

	Failure Rate (/5)				Traversal Time (s)			
Scenarios	TRAL	ECGM	BADGR	**Ours**	TRAL	ECGM	BADGR	**Ours**
MT: Obscured RGB sensor	1	1	**0**	**0**	21.49	19.98	19.52	**16.08**
MT: Obscured LiDAR sensor	1	1	1	**0**	18.216	17.77	18.16	**16.39**
MT: Obscured RGB and LiDAR sensor	4	2	2	**1**	32.65	31.92	17.99	**17.11**
Forest: Obscured RGB sensor	2	2	0	0	18.71	17.23	18.15	**15.41**
Forest: Obscured LiDAR sensor	3	1	4	**1**	16.58	15.78	17.46	**15.96**
Forest: Obscured RGB and LiDAR sensor	5	4	4	**1**	-	25.68	17.57	**16.49**

6.5.2 Navigating with Obscured Sensors

In this set of experiments, we evaluate the robot over the mixed terrain and the forest terrain under sensor obscuration. Specifically, we use three different scenarios of sensor obscuration: i) obscured RGB sensor, ii) obscured LiDAR sensor, and iii) obscured RGB and LiDAR sensor. We introduce obscuration in the color camera by partially covering the sensor with mud, whereas for a

LiDAR sensor, we obscure a substantial field of view using tree branches as seen in Figure 6.4. These scenarios are common with everyday unstructured terrains. Our approach is trained on data collected while the robot is manually controlled by an expert to traverse over various simple terrains such as grass, sand, gravel, and rocks. Then the learned model is deployed with sensor obscurations and the evaluation metrics for each of the methods are computed across three runs over each of the terrains.

The qualitative results obtained by our approach and its comparison with other methods are presented in Table 6.1. In terms of the failure rate metric, we observe that both BADGR and our approach obtain the lowest values, whereas the methods of TRAL and ECGM have high failures. Navigation failures in mixed terrain most commonly occurred when transitioning from mud to large rocks or because the robot was stuck in tall grass. Specifically, failures of transitioning from mud to large rocks are observed with the TRAL and the ECGM approach, whereas the failure of getting stuck in tall grass is observed with the BADGR approach. In addition, with respect to the different scenarios of sensor obscuration, we observe that approaches have a lower failure rate when only RGB sensor is obscured but when the LiDAR sensor is obscured the failure rates increase, especially with the BADGR approach. Across all methods, most failures are observed when both the RGB and LiDAR data are obscured.

Navigation failures in forest terrain occur either when robots are stuck in obstacles (i.e., large trees), get high centered by logs occluded by tall grass and snow, or due to wheel slip in snow or grass. Distinctively, RGB sensor obscuring leads to failures of wheel slip, whereas, LiDAR obscuration leads to methods getting stuck in obstacles, especially in the direction where the LiDAR's field of view is obscured. Similar to mixed terrain, we observe higher failure rates in forest terrain when LiDAR sensors are obscured as compared to the RGB sensors. Also, in general, we observe that methods have higher failure rates when traversing in forest terrain as compared to the mixed terrain, with increased failures caused due to obstacles in the terrain. Overall, we observe that our approach is robust to the sensor obscurations and executes navigational behaviors indifferently, thus performing consistently across the different scenarios and achieving the least failures.

We calculate the traversal time metric by averaging their values across all successful runs. In terms of the traversal time metric, we observe that both the TRAL and ECGM approaches have more variations in traversal time, which vary according to the sensor being obscured. Specifically, in both TRAL and ECGM approaches, it is observed that traversal time increases to a larger extent when obscuring the RGB sensor as compared to the LiDAR sensor with the highest traversal time recorded when both the RGB and the LiDAR sensor fails. This is because both the TRAL and ECGM approaches mainly rely on features extracted from both the RGB and LiDAR sensors for generating navigational behaviors. When sensors are obscured, the features generated from obscured regions of the observation are zeros. From [6] and [34], we learn that RGB sensors, as compared to LiDAR, have a higher impact on generated navigational behaviors, which fits our observations. Comparatively, BADGR has a consistent traversal time across the different scenarios, with traversal time decreasing with obscuring the LiDAR sensor. This is observed as BADGR uses LiDAR scans to detect obstacles and steers away when near tall grass or potential obstacles. However, with obscured LiDAR it steers less, leading to lower traversal time but increase failures. Finally, our approach can use modality-invariant representations which are not affected by sensor obscurations and traverse with consistent traversal time across all the scenarios and also achieve the lowest traversal time metric.

Table 6.2 Quantitative results for scenarios when the robot encounters sensory failures. Successful runs (with no navigation failures) are used to calculate the metrics of traversal time and adaptation time.

	Failure Rate (/5)				Traversal Time (s)			
Scenarios	TRAL	ECGM	BADGR	**Ours**	TRAL	ECGM	BADGR	**Ours**
MT: Failed RGB sensor	2	2	**0**	1	32.27	29.53	20.36	**17.16**
MT: Failed LiDAR sensor	2	**1**	2	**1**	25.49	23.93	18.03	**17.09**
MT: Failed RGB and LiDAR sensor	5	5	4	**2**	-	-	**17.52**	17.89
Forest: Failed RGB sensor	4	3	**0**	**0**	29.93	27.50	19.63	**17.31**
Forest: Failed LiDAR sensor	5	3	3	**1**	-	25.24	18.67	**16.58**
Forest: Failed RGB and LiDAR sensor	5	5	5	**3**	-	-	-	**16.11**

6.5.3 Navigating with Failed Sensors

In this set of experiments, we evaluate the robot over the mixed terrain and the forest terrain under simulated sensor failure. Specifically, we use three different scenarios: i) failed RGB sensor, ii) failed LiDAR sensor and iii) failed RGB and LiDAR sensors. Each of these failures is simulated in runtime. Specifically, failure in RGB sensors is generated by providing a zero image (black image) to the different methods and failure in LiDAR is simulated by using zero values for all LiDAR point values during runtime. These failures represent the most common sensory challenges that arise while traversing real-world terrains. We use the learned model from previous experiments with no additional training and test it in the two terrains with sensor failures. The evaluation metrics for each of the methods are computed across three runs over each of the terrains.

The qualitative results obtained by our approach and its comparison with other methods for the different scenarios of sensor failure are presented in Table 6.2. We observe a similar trend of failures as both our approach and BADGR obtain the lowest values of failures, whereas both TRAL and ECGM have higher failures. However, all the approaches have a higher failure rate when navigating with failed sensors. Specifically, we observe that the TRAL and ECGM approach has no successful runs in either mixed or forest terrain when both the RGB and LiDAR sensors fail. These failures are both due to collision with obstacles or wheel slips. We observe that in forest terrain with no RGB and LiDAR sensor, the BADGR approach has no successful runs with all the failures occurring due to collisions. Comparatively, our approach has at least one successful run in all the terrains while performing better in all the scenarios.

Similar to previous experiments, we calculate the traversal time metric by averaging their values across all successful runs. With the scenario of failed sensors, we observe that traversal time increases significantly for both the TRAL and ECGM approaches and we observe a similar trend to previous experiments where traversal time increases to a larger extent when obscuring RGB sensor as compared to the LiDAR sensor with the highest traversal time recorded when both the RGB and the LiDAR sensor fails. Thus failing both sensors lead to a very slow traversal in TRAL and ECGM approach. However, both BADGR and our approach have a consistent traversal time across the different scenarios. We do observe that in all successful runs where the LiDAR sensor fails the traversal time improves for the BADGR approach similar to the previous experiments but causes more failures as there are no successful runs in the forest scenarios when both the sensors fail. Our approach through generating modality-invariant representation from working sensory modalities is able to traverse even with sensor failures and achieves the lowest traversal time in most of the scenarios.

6.6 Conclusion

In this chapter, we introduce a novel human-inspired approach to learning capability-aware perceptual adaptation, which allows the robot to adjust to perceptual challenges and successfully navigate in unstructured off-road environments. Our approach learns capability awareness in perception through the learning of a modality invariant trajectory representation that allows for encoding a common representation of information from all the sensors in a much smaller observation space in a self-supervised fashion. To evaluate our approach, we perform extensive experiments using a physical ground robot with real-world scenarios of different perceptual challenges over various unstructured off-road terrains. Experimental results prove that our method can better adapt to these perceptual challenges when navigating over unstructured terrains and outperform previous and baseline methods.

CHAPTER 7

ENHANCING CONSISTENT GROUND MANEUVERABILITY BY ROBOT ADAPTATION TO COMPLEX OFF ROAD TERRAINS

This chapter is modified from a conference paper published at the 2021 Conference on Robot Learning (CoRL)[19]. This work was partially supported by the NSF CAREER Award IIS-1942056, NSF CNS-1823245, and ARL SARA Program W911NF-20-2-0107.

Sriram Siva[20], Maggie Wigness[21] John Rogers[22] and Hao Zhang[23].

7.1 Abstract

Terrain adaptation is a critical ability for a ground robot to effectively traverse unstructured off-road terrain in real-world field environments such as forests. However, the expected or planned maneuvering behaviors cannot always be accurately executed due to setbacks such as reduced tire pressure. This inconsistency negatively affects the robot's ground maneuverability, and can cause slower traversal time or errors in localization. To address this shortcoming, we propose a novel method for consistent behavior generation that enables a ground robot's actual behaviors to more accurately match expected behaviors while adapting to a variety of complex off-road terrains. Our method learns offset behaviors in a self-supervised fashion to compensate for the inconsistency between the actual and expected behaviors without requiring the explicit modeling of various setbacks. To evaluate the method, we perform extensive experiments using a physical ground robot over diverse complex off-road terrain in real-world field environments. Experimental results show that our method enables a robot to improve its ground maneuverability on complex unstructured

[19] ©2021 CoRL. Reprinted with permissions from Sriram Siva, Maggie Wigness, John Rogers, and Hao Zhang. ”Enhancing Consistent Ground Maneuverability by Robot Adaptation to Complex Off Road Terrains”, in *Conference on Robot Learning* (CoRL), 2021.

[20] Primary researcher and author, Graduate Student, Colorado School of Mines

[21] Senior Research Scientist, United States DEVCOM Army Research Lab

[22] Senior Research Scientist, United States DEVCOM Army Research Lab

[23] Associate Professor, Colorado School of Mines

off-road terrain with more navigational behavior consistency, and outperforms previous and baseline methods, particularly so on challenging terrain such as that which is seen in forests.

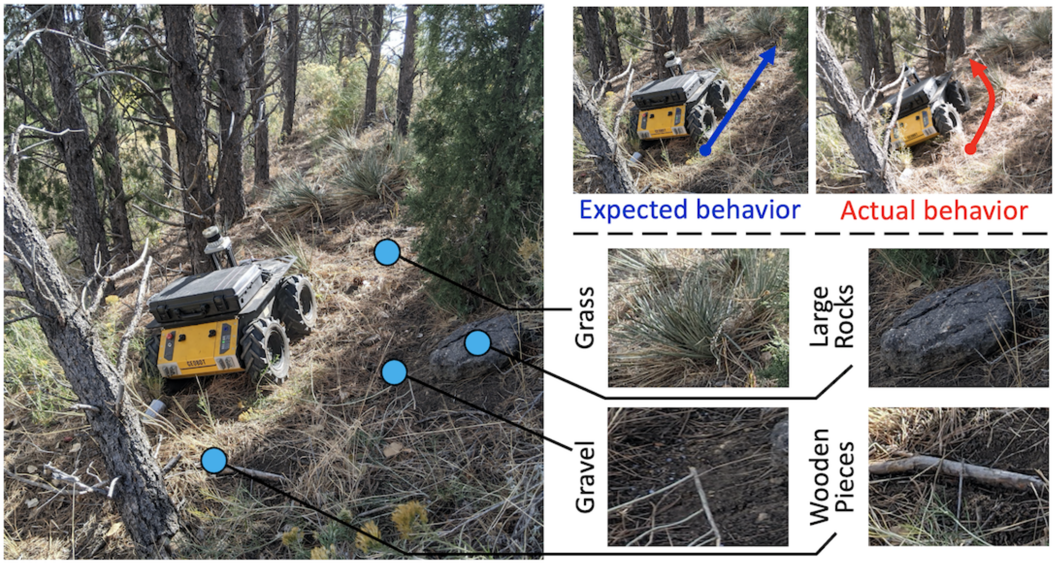


Figure 7.1 Off-road environments such as forests are unstructured and exhibit a variety of characteristics, including changing terrain types and slopes. When ground robots are deployed in these environments, their actual behaviors often do not match the expected behaviors, e.g., due to wheel slip. The inconsistency often causes slower traversal time or errors in robot state estimation. Therefore, the capability of consistent behavior generation is essential for maneuverability while ground robots navigate over unstructured off-road terrains.

7.2 Introduction

Over the past several years, autonomous ground robots have been increasingly deployed in off-road field environments to address real-world applications, including disaster response, homeland defense, and planetary exploration [158, 263, 264]. Field environments are challenging for ground robots to navigate over because the terrain is unstructured and cannot be fully modeled beforehand, as depicted in Figure 7.1. Terrain adaptation, the robot's ability to adjust its behaviors to perceived terrains, is therefore an essential ability to traverse over unstructured terrains [109, 159].

Given its importance, robot terrain adaptation has recently been widely investigated. Previous learning-based methods can be divided into two broad categories: terrain classification and terrain adaptation. The first category uses a robot's exteroceptive and proprioceptive sensory data to classify terrain types and estimate traversability for robot navigation over the terrain [111, 118, 265–267]. This category also includes techniques that model terrain complexity for navigation planning [109, 268]. The second category of methods focus on directly generating adaptive behaviors according to terrain in order to successfully complete navigation tasks [36, 109, 117, 269]. Specifically, learning from demonstration (LfD) is widely used to transfer human expertise to robots in order to achieve human-level robot navigational control [6, 117, 154].

However, the expected navigational behaviors generated by previous approaches cannot always be executed accurately by a ground robot when traversing over unstructured off-road terrain, i.e., the actual and expected behaviors may not be consistent. This inconsistency is mainly caused by setbacks [270, 271] that are defined as factors that increase the difficulty for a ground robot to achieve its expected behaviors. Example setbacks include wheel slip, reduced tire pressure, and heavy payload. Existing learning-based methods for robot navigation generally ignore these setbacks, which often leads to the robot not being able to consistently execute the learned behaviors. The challenge of how to learn consistent navigational behaviors in a self-supervised fashion has not been well addressed.

To address this shortcoming, we develop an approach for consistent navigational behavior generation. Our approach learns offset behaviors in a self-supervised fashion, allowing the robot to compensate for the inconsistency between the actual and expected behaviors without explicitly modeling the setbacks, while also adaptively navigating over changing terrain. In addition, our method is able to integrate multi-modal features to characterize terrain and estimate the importance of features to enable terrain-aware ground navigation. This is all implemented in a unified regularized optimization framework with a theoretical convergence guarantee.

The key novelty of this chapter is the introduction of a method to enhance consistent ground maneuverability, which advances the state-of-the-art by enabling a ground robot's actual behaviors to accurately match its expected behaviors while adapting to a variety of complex unstructured off-road terrain. The specific novel contributions include:

- We propose a novel mathematical formulation to generate consistent navigational behaviors by learning offset behaviors in a self-supervised fashion. We also introduce new regularization terms to learn important terrain features from multi-sensory observations and fuse them together to improve robustness of robot adaptation to unstructured terrain.
- We propose a new optimization algorithm to address the formulated regularized optimization problem with dependent variables and non-term regularization terms, which holds a theoretical guarantee to effectively converge to an optimal solution.

As an experimental contribution, we provide a comprehensive performance evaluation of learning-based terrain adaptation methods by designing a set of robot navigation scenarios over a wide variety of individual and complex unstructured off-road terrains.

7.3 Related Work

Related research on robot terrain adaptation can be broadly classified into two categories, including terrain classification and robot adaptation.

Terrain classification methods use sensory data from a robot to classify the terrain. Many earlier methods were developed to address the specific needs of larger vehicles [247, 272] and the classification process was typically performed in a manual or pre-selected fashion. Some methods used a pre-existing terrain map and terrain ruggedness data to achieve high-speed terrain navigation [111, 118, 249, 266, 267, 273]. Learning-based methods are commonly used to classify terrain for navigation. Color-based terrain classification was performed to generate navigational behaviors by labeling obstacles [133]. More recently, semantic segmentation neural network architectures [250] have been successfully used to classify off-road terrain [252, 274] and perform robot navigation tasks accordingly. However, these methods rely on a discrete categorization of terrain types. The methods typically do not characterize complex unstructured off-road terrain well and cannot directly enable robot adaptation in real-world environments, where environments have a wide variety of characteristics (as seen in Figure 7.1).

Robot adaptation methods focus on enabling robots to intelligently adapt to various unstructured terrain. The general problem of robot adaptation is commonly studied in robotics [58, 83, 89, 116, 275, 276] using high-level behavior models. Earlier works considered ground speed as the optimization variable and formulated a method for trading progress and velocity with changing terrain characteristics [146, 147]. Learning-based methods for terrain adaptation have gained attention because of their effectiveness and flexibility [277]. Early work addressed terrain adaptation from the perspective of online learning that updates model parameters in the execution phase [66, 278, 279]. However, these methods lacked the ability to quickly adapt to sudden terrain changes. Accordingly, methods were developed that generate navigational behaviors using open-loop controllers according to the predicted terrain characteristics [139]. Methods to mimic expert controls were also developed aiming to achieve human-level maneuverability [6, 117]. Recently, navigational affordances are learned from experts for off-road navigation [164]. External disturbances were also considered in the learning-based methods [280–282]. However, these methods often require modeling of robot dynamics or explicit human demonstrations, and cannot generate consistent ground navigation that adapt to diverse terrains.

7.4 Approach

Notation: We denote scalars using lowercase italic letters (e.g., $m \in \mathbb{R}$), vectors using boldface lowercase letters (e.g., $\mathbf{m} \in \mathbb{R}^p$), matrices using boldface capital letters, e.g., $\mathbf{M} = \{m_j^i\} \in \mathbb{R}^{p \times q}$ with its i-th row and j-th column denoted as $\mathbf{m}^i$ and $\mathbf{m}_j$, respectively. We use boldface capital Euler script letters to denote tensors (i.e., 3D matrices), e.g., $\boldsymbol{\mathcal{M}} = \{m_j^{i(k)}\} \in \mathbb{R}^{p \times q \times r}$. Unstacking tensor $\boldsymbol{\mathcal{M}}$ along its height (p), width (q) and depth (r) provides slices of matrices $\mathbf{M}^i \in \mathbb{R}^{q \times r}$, $\mathbf{M}_j \in \mathbb{R}^{p \times r}$ and $\mathbf{M}^{(k)} \in \mathbb{R}^{p \times q}$, respectively [283].

7.4.1 Problem Formulation for Terrain-Aware Navigation

As a robot traverses over terrain, at each time step, we extract multi-modal features from observations acquired from multiple sensors installed on the robot including visual camera, LiDAR, and IMU. We concatenate all features extracted at time step t into a vector and denote it as

$\mathbf{x}^{(t)} \in \mathbb{R}^d$, where $d = \sum_{j=1}^{m} d^j$ and d^j is the dimensionality of the j-th feature modality with m as the number of modalities. We stack features extracted from a sequence of consecutive c time steps into a matrix and represent it as a terrain feature instance denoted as $\mathbf{X} = [\mathbf{x}^{(t)}; \ldots; \mathbf{x}^{(t-c)}] \in \mathbb{R}^{d \times c}$. We further denote the set of n feature instances that are obtained as a robot traverses over various terrains, and denote this set as a terrain feature tensor $\boldsymbol{\mathcal{X}} = [\mathbf{X}_1, \ldots, \mathbf{X}_n] \in \mathbb{R}^{d \times n \times c}$.

We use $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n] \in \mathbb{R}^{r \times n}$ to denote the robot's expected navigational behaviors associated with $\boldsymbol{\mathcal{X}}$, where $\mathbf{y}_i \in \mathbb{R}^r$ is a vector of r behaviors corresponding to $\mathbf{X}_i$. Behaviors are encoded by the control variables (e.g., linear and angular velocities) that decide the robot's motion at the present time t. Due to momentum, a robot often has continuous motion and observations; thus, considering a history of past c observations can provide more information to generate navigational behaviors at the present time. Accordingly, we estimate the robot's behaviors $\mathbf{y}_i$ using $\mathbf{X}_i = [\mathbf{x}_i^{(t)}; \ldots; \mathbf{x}_i^{(t-c)}]$, taking into account the history of c observations. Then, the problem of navigational behavior generation can be formulated as:

$$\min_{\boldsymbol{\mathcal{W}}} \|\boldsymbol{\mathcal{W}} \otimes_3 \boldsymbol{\mathcal{X}} - \mathbf{Y}\|_F^2 + \lambda_1 \|\boldsymbol{\mathcal{W}}\|_M \tag{7.1}$$

where $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{d \times r \times c}$ is a weight tensor used to encode the importance of each element in $\boldsymbol{\mathcal{X}}$ towards estimating navigational behaviors. Each tensor element $w_j^{i(k)} \in \boldsymbol{\mathcal{W}}$ denotes the weight of the i-th terrain feature from the k-th past time step to recognize the j-th behavior type. The operator $\otimes$ denotes the tensor product and $\otimes_3$ is defined as the tensor product that performs the sum of mode-3 multiplication [283] between feature tensor $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{W}}$. In Eq. (7.1), the tensor product $\otimes_3$ takes each terrain feature instance $\mathbf{X}_i \in \boldsymbol{\mathcal{X}}$, and multiplies it with the weight tensor $\boldsymbol{\mathcal{W}}$.

The first term in Eq. (7.1) is a loss function that encodes the error of using terrain features in $\boldsymbol{\mathcal{X}}$ to estimate the robot behaviors, through the learning model parameterized by $\boldsymbol{\mathcal{W}}$. Furthermore, our loss function encodes the non-linear nature of robot navigational behavior generation as a linear function of terrain features $\boldsymbol{\mathcal{X}}$. This can be achieved because: i) for short periods of time (i.e., $c = [1, 30]$), the dynamics of the robot do not dramatically change [284, 285] and thus the non-linearity in the robot is not severe, and ii) with high-dimensional features (e.g., $d > 10000$), learning non-linear tasks can be lifted to a linear space [187, 286]. The second term in Eq. (7.1) is a regularization term named the feature modality norm and is mathematically defined as:

$$\|\boldsymbol{\mathcal{W}}\|_M = \sum_{i=1}^{m} \|\mathbf{W}^i\|_F = \sum_{i=1}^{m} \sqrt{\sum_{j=1}^{r}\sum_{k=1}^{c} (\mathbf{w}_j^{i(k)})^\top (\mathbf{w}_j^{i(k)})} \tag{7.2}$$

where $\|.\|_F$ is the Frobenius norm and $\mathbf{W}^i \in \mathbb{R}^{b \times c}$ is the slice of the matrix obtained by unstacking the weight tensor $\boldsymbol{\mathcal{W}}$ along its height d. The feature modality norm groups together weights within a feature modality and enforces sparsity among different modalities, thus, identifying the most descriptive features for behavior generation. This is a critical capability for ground robot navigation since different features typically capture different characteristics of the unstructured terrains (e.g., color, slope, and roughness), and have different effects toward generating navigational behaviors. The trade-off hyperparameter λ_1 in Eq. (7.1) is used to balance the loss and the regularization term.

The problem formulation in Eq. (7.1) allows ground robots to adapt their navigational behaviors according to different terrain features. However, due to setbacks that reduce the effectiveness of robot navigation, such as wheel slip, heavy payload, and reduced tire pressure [248], the robot's actual behaviors may not match the expected behaviors.

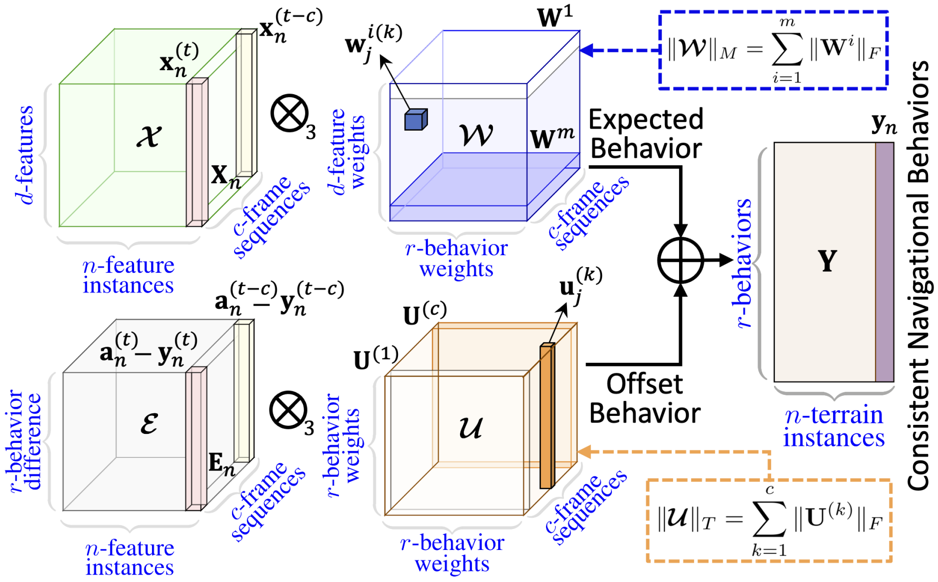


Figure 7.2 Overview of our proposed approach.

7.4.2 Consistent Navigational Behavior Generation

The key novelty focuses on a principled method for ground robots to generate consistent navigational behaviors that adapt to unstructured terrain. As illustrated by Figure 7.2, besides generating terrain-aware expected behaviors, our method monitors the difference between the actual and expected navigational behaviors caused by setbacks, and computes an offset to reduce the difference. This allows our approach to achieve consistent robot behaviors without the requirement of explicitly modeling all of the setbacks.

Mathematically, we denote the actual behaviors as $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n] \in \mathbb{R}^{r \times n}$, where $\mathbf{a}_i \in \mathbb{R}^r$ denotes the actual behaviors executed by the robot when observing the i-th terrain feature instance $\mathbf{X}_i$. The actual robot behaviors can be estimated using a pose estimation technique, e.g., based on SLAM or visual odometry [287]. Setbacks can cause the robot's actual behaviors to not match its expected behaviors. This difference in navigational behaviors over the past c-time steps is computed as $\mathbf{E} = [(\mathbf{a}^{(t)} - \mathbf{y}^{(t)}); \dots; (\mathbf{a}^{(t-c)} - \mathbf{y}^{(t-c)})] \in \mathbb{R}^{r \times c}$ and the behavior differences for all the instances in $\boldsymbol{\mathcal{X}}$ is denoted as a behavior difference tensor $\boldsymbol{\mathcal{E}} = [\mathbf{E}_1, \dots, \mathbf{E}_n] \in \mathbb{R}^{r \times n \times c}$.

We then introduce a loss function to encode consistent behavior generation as follows:

$$\min_{\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{W}}} \|\boldsymbol{\mathcal{W}} \otimes_3 \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{U}} \otimes_3 \boldsymbol{\mathcal{E}} - \mathbf{A}\|_F^2 + \lambda_1 \|\boldsymbol{\mathcal{W}}\|_M \tag{7.3}$$

where $\boldsymbol{\mathcal{U}} = [\mathbf{U}^1, \dots, \mathbf{U}^r] \in \mathbb{R}^{r \times r \times c}$ is a weight tensor with $\mathbf{U}^j \in \mathbb{R}^{r \times c}$ indicates the importance of behavior differences $\mathbf{E}$ towards generating the j-th offset behaviors. Using data from a history of c time steps allows our method to consider inertia (i.e., resistance to change in behaviors) during navigation. Mathematically, this implies that $\mathbf{U}^{(k)}, k = 1, \dots, c$; has non-zero elements as opposed to each element in $\mathbf{U}^{(k)}$ being 1 at $k = 1$ and 0 at $k \neq 1$ when robot inertia is not considered.

The loss function in Eq. (7.3) models the actual behavior by considering both terrain features $\boldsymbol{\mathcal{X}}$ and the behavior differences $\boldsymbol{\mathcal{E}}$ to achieve consistent navigational behaviors. Because of inertia, historical data from different past time steps may contribute differently towards generating offset behaviors (e.g., a heavier robot with larger inertia often needs to consider a longer history). Thus, we propose a new regularization term to explore which time steps in the historical data are more important for generating the offset behaviors. We name this regularization term the temporal norm, which is expressed by:

$$\|\boldsymbol{\mathcal{U}}\|_T = \sum_{k=1}^{c} \|\mathbf{U}^{(k)}\|_F = \sum_{k=1}^{c} \sqrt{\sum_{j=1}^{r} \|\mathbf{u}_j^{(k)}\|_2^2} \tag{7.4}$$

where $\mathbf{U}^{(k)} = [\mathbf{u}_1^{(k)}, \dots, \mathbf{u}_r^{(k)}] \in \mathbb{R}^{r \times r}$ is the weight matrix, and $\mathbf{u}_j^{(k)}$ indicates the importance of the j-th behavior difference $\mathbf{a}_j^{(k)} - \mathbf{y}_j^{(k)}$ from k-the past time step towards generating offset behaviors. This norm groups together weights for the vector of behavior differences at each time step, and enforces sparsity between weights at different time steps to identify the most important time steps.

Using both norms to generate consistent robot navigational behaviors while identifying important feature modalities and historical time steps, the final objective function becomes:

$$\min_{\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{W}}} \|\boldsymbol{\mathcal{W}} \otimes_3 \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{U}} \otimes_3 \boldsymbol{\mathcal{E}} - \mathbf{A}\|_F^2 + \lambda_1 \|\boldsymbol{\mathcal{W}}\|_M + \lambda_2 \|\boldsymbol{\mathcal{U}}\|_T \tag{7.5}$$

where $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ are trade-off hyper-parameters.

After computing the optimal values of the weight tensors $\boldsymbol{\mathcal{W}}$ and $\boldsymbol{\mathcal{U}}$ according to Algorithm 7, in the training phase, a robot can apply our self-reflective terrain-aware adaptation method to generate consistent navigational behaviors during execution. At each time step t_0 in the execution phase, the robot extracts multi-modal features $\mathbf{X}_{t_0}$ from observations obtained from its on-board sensors over the past c-time steps. Then, the robot also estimates the corresponding actual behaviors (measured using a pose estimation technique based upon SLAM or visual odometry [287]) and computes the matrix of behavior differences at time t_0 as $\mathbf{E}_{t_0}$. Then our approach can be used by the robot to generate consistent actual navigational behaviors as:

$$\mathbf{y} = \boldsymbol{\mathcal{W}} \otimes_3 \mathbf{X}_{t_0} + \boldsymbol{\mathcal{U}} \otimes_3 \mathbf{E}_{t_0} \tag{7.6}$$

The first term in Eq. (7.6) generates the expected navigational behaviors according to the terrain it traverses, which allows a ground robot to adapt its navigational behaviors to unstructured terrains. The second term in Eq. (7.6) provides offset behaviors based on monitoring the difference between actual and expected behaviors in order to compensate for the setbacks.

Algorithm 6: Our algorithm to solve the regularized optimization problem in Eq. (7.5).

```
Input  : $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d\times n\times c}$, $\mathbf{A} \in \mathbb{R}^{r\times n}$, and $\boldsymbol{\mathcal{E}} \in \mathbb{R}^{r\times n\times c}$
Output : The weight tensors $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{d\times r\times c}$ and $\boldsymbol{\mathcal{U}} \in \mathbb{R}^{r\times r\times c}$
Initialize $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{d\times r\times c}$ and $\boldsymbol{\mathcal{U}} \in \mathbb{R}^{r\times r\times c}$;
while not converge do
    Calculate each block diagonal matrix $\mathbf{Q}^{(k)}$ with $i$-th diagonal block given as $\frac{1}{2\|\mathbf{W}^{i(k)}\|_F}\mathbf{I}_{d^i}$;
    Compute the matrices $\mathbf{W}^{(k)}$ according to Eq. (7.8);
    Calculate the block diagonal matrix $\mathbf{P}$ with the $k$-th diagonal block as $\frac{1}{2\|\mathbf{U}^{(k)}\|_F}\mathbf{I}_r$;
    Compute the matrices $\mathbf{U}^{(k)}$ according to Eq. (7.10);
return: $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{d\times r\times c}$ and $\boldsymbol{\mathcal{U}} \in \mathbb{R}^{r\times r\times c}$
```

7.4.3 Optimization Algorithm

The optimization problem in Eq. (7.5) is challenging to solve because the regularization terms are not smooth and because the objective function includes dependent variables. Thus, we derive a new iterative optimization algorithm to obtain the optimal solution to Eq. (7.5). This algorithm is shown in Algorithm 7, which provides an alternating minimization algorithm that alternatively updates the parameter tensors in each iteration until convergence.

To solve for the optimal weight tensor $\boldsymbol{\mathcal{W}}$, we minimize Eq. (7.5) with respect to $\mathbf{W}^{(k)}$ resulting in:

$$2\mathbf{X}^{(k)}(\mathbf{X}^{(k)})^\top\mathbf{W}^{(k)} - 2\boldsymbol{\mathcal{X}} \otimes_3 \mathbf{A} + 2(\boldsymbol{\mathcal{U}} \otimes_3 \boldsymbol{\mathcal{E}})^\top \otimes_3 \boldsymbol{\mathcal{X}} + \lambda_1\mathbf{Q}^{(k)}\mathbf{W}^{(k)} = 0 \tag{7.7}$$

where each $\mathbf{Q}^{(k)} \in \mathbb{R}^{d_i\times d_i}$ is a diagonal matrix with the i-th diagonal block computed by $\frac{1}{2\|\mathbf{W}^{i(k)}\|_F}\mathbf{I}_{d^i}$ and $\mathbf{I}_{d^i}$ is an identity matrix. Then, we compute each $\mathbf{W}^{(k)}$ as:

$$\mathbf{W}^{(k)} = \big(2\mathbf{X}^{(k)}(\mathbf{X}^{(k)})^\top + \lambda_1\mathbf{Q}^{(k)}\big)^{-1}\big(2\boldsymbol{\mathcal{X}} \otimes_3 \mathbf{A} - 2(\boldsymbol{\mathcal{U}} \otimes_3 \boldsymbol{\mathcal{E}})^\top \otimes_3 \boldsymbol{\mathcal{X}}\big) \tag{7.8}$$

Because each of the block-diagonal matrices $\mathbf{Q}^{(k)}$ are dependent on $\boldsymbol{\mathcal{W}}$ and also each slice of matrix $\mathbf{W}^{(k)}$ is dependent on corresponding $\mathbf{Q}^{(k)}$, an iterative algorithm is required to compute them.

To compute the optimal $\boldsymbol{\mathcal{U}}$, we calculate the derivative of the objective function in Eq. (7.5) with respect to $\mathbf{U}^{(k)}$ and set the equation to zero as:

$$2\mathbf{E}^{(k)}(\mathbf{E}^{(k)})^\top\mathbf{U}^{(k)} - 2\mathbf{A} \otimes_3 \boldsymbol{\mathcal{E}} + 2(\boldsymbol{\mathcal{W}} \otimes_3 \boldsymbol{\mathcal{X}})^\top \otimes \boldsymbol{\mathcal{E}} + \lambda_2\mathbf{P}\mathbf{U}^{(k)} = 0 \tag{7.9}$$

where $\mathbf{P} \in \mathbb{R}^{r \times r}$ is a diagonal matrix with the k-th diagonal block as $\frac{1}{2\|\mathbf{U}^{(k)}\|_F}\mathbf{I}_r$, with $\mathbf{I}_r$ being an identity matrix. Then, we compute each $\mathbf{U}^{(k)}$ in a closed-form solution as:

$$\mathbf{U}^{(k)} = \big(2\mathbf{E}^{(k)}(\mathbf{E}^{(k)})^\top + \lambda_2 \mathbf{P}\big)^{-1}\big(2\mathbf{A} \otimes_3 \boldsymbol{\mathcal{E}} - 2(\boldsymbol{\mathcal{W}} \otimes_3 \boldsymbol{\mathcal{X}})^\top \otimes \boldsymbol{\mathcal{E}}\big) \tag{7.10}$$

This $\boldsymbol{\mathcal{U}}$ is then used to calculate $\boldsymbol{\mathcal{W}}$ in the next iteration. Similar to $\mathbf{Q}^{(k)}$ and $\boldsymbol{\mathcal{W}}$, $\mathbf{P}$ and $\boldsymbol{\mathcal{U}}$ are interdependent. Thus, we develop an iterative algorithm to solve the formulated optimization problem, which is described in Algorithm 7.

7.5 Proof of Convergence for the Proposed Optimization Algorithm

In this section, we prove that Algorithm 3 decreases the value of the objective function in Eq. (7.5) in each iteration, and the algorithm converges to an optimal solution. Our optimization algorithm follows the iterative re-weighted method [55] for optimizing objectives with non-smooth regularization terms.

First, we present a lemma:

Lemma 5: Given any two vectors $\mathbf{a}$ and $\mathbf{b}$, the following inequality holds:

$$\|\mathbf{b}\|_2 - \frac{\|\mathbf{b}\|_2^2}{2\|\mathbf{a}\|_2} \leq \|\mathbf{a}\|_2 - \frac{\|\mathbf{a}\|_2^2}{2\|\mathbf{a}\|_2}$$

Proof.

$$-(\|\mathbf{b}\|_2 - \|\mathbf{a}\|_2)^2 \leq 0$$

$$-\|\mathbf{b}\|_2^2 - \|\mathbf{a}\|_2^2 + 2\|\mathbf{b}\|_2\|\mathbf{a}\|_2 \leq 0$$

$$2\|\mathbf{b}\|_2\|\mathbf{a}\|_2 - \|\mathbf{b}\|_2^2 \leq \|\mathbf{a}\|_2^2$$

$$\|\mathbf{b}\|_2 - \frac{\|\mathbf{b}\|_2^2}{2\|\mathbf{a}\|_2} \leq \|\mathbf{a}\|_2 - \frac{\|\mathbf{a}\|_2^2}{2\|\mathbf{a}\|_2}$$

□

From Lemma 6, we can derive the following corollary:

Corollary 4: Given any two matrices $\mathbf{A}$ and $\mathbf{B}$, the following inequality holds:

$$\|\mathbf{B}\|_F - \frac{\|\mathbf{B}\|_F^2}{2\|\mathbf{A}\|_F} \leq \|\mathbf{A}\|_F - \frac{\|\mathbf{A}\|_F^2}{2\|\mathbf{A}\|_F}$$

Theorem 5: Algorithm 1 (in the main paper) converges to an optimal solution to the formulated regularized optimization problem in Eq. (5) (in the main paper).

Proof. According to Step 4 of Algorithm 1, in the s-th iteration, the value of $\mathbf{W}^{(k)}(s+1)$ can be calculated as:

$$\mathbf{W}^{(k)}(s+1) = \|\boldsymbol{\mathcal{W}}(s) \otimes_3 \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{U}}(s) \otimes_3 \boldsymbol{\mathcal{E}} - \mathbf{A}\|_F^2 + \lambda_1 Tr(\mathbf{W}^{(k)}(s))^\top \mathbf{Q}^{(k)}(s)\mathbf{W}^{(k)}(s) \tag{7.11}$$

where the operator Tr denotes the trace of a matrix and $\mathbf{Q}^{(k)}(s+1) = \frac{1}{2\|\mathbf{W}^{(k)}(s)\|_F}\mathbf{I}_{d^i}$.

From Step 6 of Algorithm 1, we obtain:

$$\mathbf{U}^i(s+1) = \|\boldsymbol{\mathcal{W}}(s+1) \otimes_3 \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{U}}(s) \otimes_3 \boldsymbol{\mathcal{E}} - \mathbf{A}\|_F^2 + \lambda_2 Tr(\mathbf{U}^{(k)}(s))^\top \mathbf{P}(s+1)\mathbf{U}^{(k)}(s) \tag{7.12}$$

where $\mathbf{P}(t+1) = \frac{1}{2\|\mathbf{U}^{(k)}\|_F}\mathbf{I}_r$.

Then, we can derive that:

$$\begin{aligned}
&\mathcal{J}(s+1) + \sum_{k=1}^{c} \lambda_1 Tr(\mathbf{W}^{(k)}(s+1))^\top \mathbf{Q}^{(k)}(s+1)\mathbf{W}^{(k)}(s+1) \\
&\leq \mathcal{J}(s) + \sum_{k=1}^{c} \lambda_1 Tr(\mathbf{W}^{(k)}(s))^\top \mathbf{Q}^{(k)}(s)\mathbf{W}^{(k)}(s)
\end{aligned} \tag{7.13}$$

where $\mathcal{J}(s) = \|\boldsymbol{\mathcal{W}}(s) \otimes_3 \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{U}}(s) \otimes_3 \boldsymbol{\mathcal{E}} - \mathbf{A}\|_F^2$.

After substituting $\mathbf{Q}^{(k)}$, we obtain:

$$\mathcal{J}(s+1) + \sum_{i=1}^{m}\sum_{k=1}^{c}(\lambda_1 \frac{\|\mathbf{W}^{i(k)}(s+1)\|_F^2}{2\|\mathbf{W}^{i(k)}(s)\|_F}) \leq \mathcal{J}(s) + \sum_{i=1}^{m}\sum_{k=1}^{c}(\lambda_1 \frac{\|\mathbf{W}^{i(k)}(s)\|_F^2}{2\|\mathbf{W}^{i(k)}(s)\|_F}) \tag{7.14}$$

From Lemma 6 and Corollary 4, $\forall k = 1, \ldots, c$, we obtain:

$$\sum_{k=1}^{c}\left(\|\mathbf{W}^{i(k)}(s+1)\|_F-\frac{\|\mathbf{W}^{i(k)}(s+1)\|_F^2}{2\|\mathbf{W}^{i(k)}(s)\|_F}\right)\leq\sum_{k=1}^{c}\left(\|\mathbf{W}^{i(k)}(s)\|_F-\frac{\|\mathbf{W}^{i(k)}(s)\|_F^2}{2\|\mathbf{W}^{i(k)}(s)\|_F}\right)\tag{7.15}$$

Adding Eqs. (7.14) and (7.15) on both sides, we obtain:

$$\mathcal{J}(s+1)+\sum_{i=1}^{m}(\lambda_1\|\mathbf{W}^i(s+1)\|_F)\leq\mathcal{J}(s)+\sum_{i=1}^{m}(\lambda_1\|\mathbf{W}^i(s)\|_F)\tag{7.16}$$

Eq. (7.16) shows that, given a fixed $\boldsymbol{\mathcal{U}}$ when updating $\boldsymbol{\mathcal{W}}$, the updated $\boldsymbol{\mathcal{W}}$ decreases the value of the objective function in each iteration.

Using the updated $\boldsymbol{\mathcal{W}}$, we can derive that:

$$\begin{aligned}&\mathcal{F}(s+1)+\sum_{k=1}^{c}(\lambda_2 Tr(\mathbf{U}^{(k)}(s+1))^\top\mathbf{P}(s+1)(\mathbf{U}^{(k)}(s+1)))\\&\leq\mathcal{F}(s)+\sum_{k=1}^{c}(\lambda_2 Tr(\mathbf{U}^{(k)}(s))^\top\mathbf{P}(s)(\mathbf{U}^{(k)}(s)))\end{aligned}\tag{7.17}$$

where $\mathcal{F}(s)=\|\boldsymbol{\mathcal{W}}(s+1)\otimes_3\boldsymbol{\mathcal{X}}+\boldsymbol{\mathcal{U}}(s)\otimes_3\boldsymbol{\mathcal{E}}-\mathbf{A}\|_F^2$. After substituting $\mathbf{P}$, we obtain:

$$\mathcal{F}(s+1)+\sum_{k=1}^{c}(\lambda_2\frac{\|\mathbf{U}^{(k)}(s+1)\|_F^2}{2\|\mathbf{Q}^{(k)}(s)\|_F})\leq\mathcal{F}(s)+\sum_{k=1}^{c}(\lambda_2\frac{\|\mathbf{U}^{(k)}(s)\|_F^2}{2\|\mathbf{U}^{(k)}(s)\|_F})\tag{7.18}$$

Similar to Eq. (7.15), from Lemma 6 and Corollary 4, $\forall k=t,\ldots,t-c$, we obtain:

$$\sum_{k=1}^{c}\left(\|\mathbf{U}^{(k)}(s+1)\|_F-\frac{\|\mathbf{U}^{(k)}(s+1)\|_F^2}{2\|\mathbf{U}^{(k)}(s)\|_F}\right)\leq\sum_{k=1}^{c}\left(\|\mathbf{U}^{(k)}(s)\|_F-\frac{\|\mathbf{U}^{(k)}(s)\|_F^2}{2\|\mathbf{U}^{(k)}(s)\|_F}\right)\tag{7.19}$$

Adding Eqs. (7.18) and Eq. (7.19) on both sides, we obtain:

$$\mathcal{F}(s+1)+\sum_{k=1}^{c}(\lambda_2\|\mathbf{U}^{(k)}(s+1)\|_F)\leq\mathcal{F}(s)+\sum_{k=1}^{c}(\lambda_2\|\mathbf{U}^{(k)}(s)\|_F)\tag{7.20}$$

Eq. (7.20) shows that, given fixed $\boldsymbol{\mathcal{W}}$ when updating $\boldsymbol{\mathcal{U}}$, the updated $\boldsymbol{\mathcal{U}}$ decreases the value of the objective function in each iteration.

Each iteration of Algorithm 1 includes both Eq. (7.16) and Eq. (7.20). By adding these equations on both sides, we obtain:

$$\begin{aligned}&\mathcal{J}(s+1)+\mathcal{F}(s+1)+\|\boldsymbol{\mathcal{W}}(s+1)\|_M+\|\boldsymbol{\mathcal{U}}(s+1)\|_T\\ &\leq \mathcal{J}(s)+\mathcal{F}(s)+\|\boldsymbol{\mathcal{W}}(s)\|_M+\|\boldsymbol{\mathcal{U}}(s)\|_T\end{aligned} \tag{7.21}$$

At the end of each iteration of Algorithm 1, the loss function $\mathcal{F}$ is always less than or equal to $\mathcal{J}$, as $\mathcal{J}$ reduces the loss value by optimizing over $\boldsymbol{\mathcal{W}}$, and $\mathcal{F}$ reduces the loss value by optimizing over both $\boldsymbol{\mathcal{W}}$ and $\boldsymbol{\mathcal{U}}$. Then we can write:

$$\begin{aligned}&\mathcal{F}(s+1)+\|\boldsymbol{\mathcal{W}}(s+1)\|_M+\|\boldsymbol{\mathcal{U}}(s+1)\|_T\\ &\leq \mathcal{J}(s+1)+\|\boldsymbol{\mathcal{W}}(s+1)\|_M+\|\boldsymbol{\mathcal{U}}(s+1)\|_T\end{aligned} \tag{7.22}$$

Similarly, as the loss function $\mathcal{J}(s+1)$ is always less than or equal to $\mathcal{F}(s)$ as $\mathcal{J}(s+1)$ reduces the value of loss further by optimizing $\boldsymbol{\mathcal{W}}(s)$, we can write:

$$\mathcal{J}(s+1)+\|\boldsymbol{\mathcal{W}}(s+1)\|_M+\|\boldsymbol{\mathcal{U}}(s)\|_T \leq \mathcal{F}(s)+\|\boldsymbol{\mathcal{W}}(s)\|_M+\|\boldsymbol{\mathcal{U}}(s)\|_T \tag{7.23}$$

From Eq. (7.21), (7.22) and (7.23), we can write the following inequality:

$$\begin{aligned}&\mathcal{F}(s+1)+\|\boldsymbol{\mathcal{W}}(s+1)\|_M+\|\boldsymbol{\mathcal{U}}(s+1)\|_T\\ &\leq \mathcal{J}(s+1)+\|\boldsymbol{\mathcal{W}}(s+1)\|_M+\|\boldsymbol{\mathcal{U}}(s+1)\|_T\\ &\leq \mathcal{J}(s+1)+\|\boldsymbol{\mathcal{W}}(s+1)\|_M+\|\boldsymbol{\mathcal{U}}(s)\|_T\\ &\leq \mathcal{F}(s)+\|\boldsymbol{\mathcal{W}}(s)\|_M+\|\boldsymbol{\mathcal{U}}(s)\|_T\end{aligned} \tag{7.24}$$

From Eq. (7.24) we have:

$$\mathcal{F}(s+1)+\|\boldsymbol{\mathcal{W}}(s+1)\|_M+\|\boldsymbol{\mathcal{U}}(s+1)\|_T \leq \mathcal{F}(s)+\|\boldsymbol{\mathcal{W}}(s)\|_M+\|\boldsymbol{\mathcal{U}}(s)\|_T \tag{7.25}$$

Eq. (7.25) shows that the value of the objective function is decreased in each iteration. Because the objective function is convex, Algorithm 3 converges to a local optimal solution to the formulated regularized optimization problem in Eq. (7.5). □

Convergence: Algorithm 7 is guaranteed to converge to a local optimal solution to the formulated regularized optimization problem in Eq. (7.5).

Complexity: As the optimization problem in Eq. (7.5) is convex, Algorithm 7 converges fast (e.g., within tens of iterations only). In each iteration, computing Steps 3 and 5 is trivial. Steps 4 and 6 can be computed by solving a system of linear equations with quadratic complexity.

7.6 Experiments

We utilize the Clearpath Husky robot in our field experiments. The robot is equipped with an Intel Reasense D435 color-depth camera and an Ouster OS1-64 LiDAR. The robot also has a variety of sensors to measure its internal states, including IMU readings, wheel odometry, motor speed, and battery status. Linear interpolation is used to get a steady 30 Hz frame rate from all sensors.

To represent unstructured terrain, we implement multiple visual features extracted from color images to describe different terrain characteristics, including Histogram of Oriented Gradients (HOG) [101] to describe the shape and Local Binary Patterns (LBP) [152] to describe texture. We also compute an elevation map [153] from LiDAR data to represent robot-centric grid-wise elevation of the terrain. During training, expected navigational behaviors are provided by remote control to navigate the robot over unstructured terrains as fast as possible while maintaining safety. In the training and execution phases, actual behaviors are estimated from LiDAR-based SLAM [287] as robot pose. We use a sequence of 15 frames (i.e., $c = 15$), and $\lambda_1 = 0.1$ and $\lambda_2 = 10$ for all experiments.

We compare our approach with several previous state-of-the-art learning-based robot navigation techniques, including Learning from Demonstration (LfD) for robot navigation [154], multi-modal LfD (MfD) [288], and Terrain Representation and Apprenticeship Learning (TRAL) [6]. To quantitatively evaluate the performance of robot navigation, we use four metrics:

- *Failure Rate (FR)*: This metric is defined as the number of times the robot fails to complete the navigation task across a set of experimental trials. If a robot flips or is stopped by a terrain obstacle, it is considered a failure. Lower values of FR indicate better performance.

- *Traversal Time (TT)*: This metric is defined as the time taken to complete the navigation task over given terrain. Smaller values of TT indicate better performance.

- *Inconsistency*: This metric is defined as the error between the expected behavior and the actual behavior in terms of robot poses (linear position and angular). Lower values of inconsistency indicate better performance.

- *Jerkiness*: This metric is defined as the average sum of the acceleration derivatives along all axes, with lower values indicating better performance. Jerkiness indicates how smooth a robot can traverse over a terrain. Because state estimation and SLAM methods (e.g., based on Kalman filters) may assume smooth robot motions, jerkiness is a useful metric.

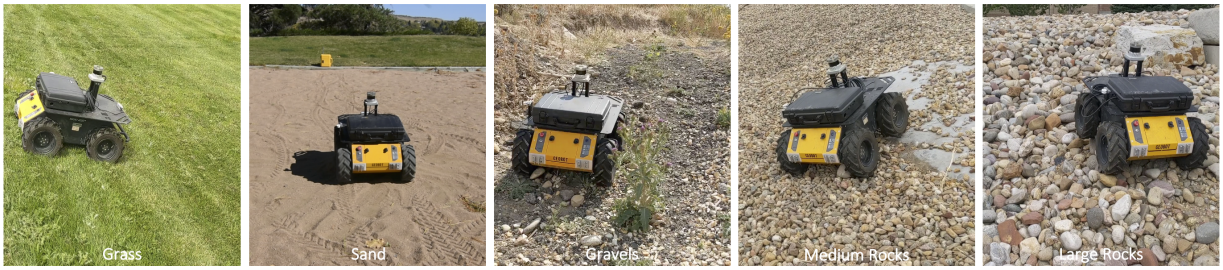


Figure 7.3 Individual types of unstructured terrain used in the experiments.

7.6.1 Navigating over Individual Types of Unstructured Terrain

In this set of experiments, the robot navigates over individual terrain using off-road tracks. Each track is made up of one type of terrain and is approximately ten meters long. Five types of terrain are used in our experiments, which are illustrated in Figure 7.3. Our approach is trained on data collected while the robot is manually controlled by an expert to traverse the terrain. Then, the learned model is deployed on the robot to autonomously navigate over the terrain. Evaluation metrics for each method are computed across ten trials on each type of individual terrain track.

The quantitative results achieved by our approach and the comparison to other methods are presented in Table 7.1. For simple individual terrains, such as grass and gravel, all methods allow the robot to successfully traverse over the terrain. However, for more challenging terrain, especially large rocks, both LfD and MfD have a high failure rate, whereas, TRAL and our approach have a low failure rate. Our approach only has one failure over the difficult large-rock terrain and outperforms other tested methods. The presented traversal time is computed by averaging the traversal time across all successful runs, i.e., it excludes the failed trials captured by the FR metric. It is observed that all methods have a similar traversal time. In successful runs, both LfD methods show less traversal time compared to other methods over rocky terrain, although they also have a much higher failure rate. Thus, the emphasis on high-speed traversal used by both the LfD methods produces an unreliable system when the robot traverses unstructured off-road terrain in real-world field environments.

Table 7.1 Quantitative results based on ten runs for scenarios when the robot traverses over *individual types of unstructured terrain*. Successful runs (with no failures) are used to calculate the metrics of traversal time, inconsistency and jerkiness.

	Failure Rate (/10)				Traversal Time (s)				Inconsistency				Jerkiness (m/s^3)			
Terrain	LfD	MfD	TRAL	**Ours**	LfD	MfD	TRAL	**Ours**	LfD	MfD	TRAL	**Ours**	LfD	MfD	TRAL	**Ours**
Grass	0	0	0	**0**	17.9	18.2	**17.4**	17.5	2.82	3.06	**1.84**	2.11	79.59	81.42	**75.18**	76.50
Sand	0	1	0	**0**	15.3	**12.7**	15.9	13.1	4.72	4.62	4.67	**4.55**	71.14	73.74	65.32	**64.22**
Gravel	0	0	0	**0**	22.9	24.1	20.4	**20.2**	4.12	4.63	3.81	**3.04**	44.95	48.27	40.26	**39.48**
M.R	1	3	0	**0**	33.2	36.9	29.4	**28.4**	7.92	9.59	4.21	**2.41**	141.48	144.22	113.34	**111.18**
L.R	6	6	2	**1**	57.8	**55.3**	63.4	60.9	24.79	28.50	9.51	**7.84**	52.55	54.30	49.50	**48.36**

Table 7.1 also presents the quantitative results for the inconsistency and jerkiness metrics. We observe that both LfD and MfD methods do not perform well and have higher values of inconsistency over individual types of terrain, especially on the large-rock terrain. TRAL has lower inconsistency and performs the best over the grass terrain. Our proposed method outperforms the previous approaches and obtains the lowest averaged inconsistency value. Finally, we also evaluate the tested methods using the jerkiness metric. An observation is that the medium-rock terrain causes the largest jerkiness measure. This is because medium-rocks terrain as compared to the large-rocks produce much more vibrations for even slow maneuvers.

7.6.2 Navigating over Complex Off-road Unstructured Terrains

In the second set of experiments, we evaluate our approach when the robot navigates over complex off-road unstructured terrain. The tracks in these experiments either show transitions between different terrain types (i.e., grass to large rocks: Gr.L.R, and grass to medium rocks: Gr.M.R) or a mixture of different terrain types in real off-road environments (i.e., Mixed Terrain I: M.T-I and Mixed Terrain II: M.T-II), as shown in Figure 7.4. In the experiments, no additional training is performed and the previously trained model from individual types of unstructured terrains is used directly.

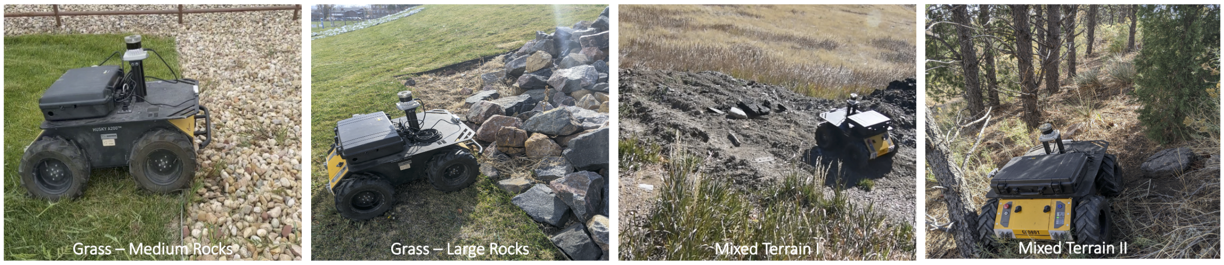


Figure 7.4 Complex unstructured off-road terrains used in the experiments.

Table 7.2 presents the quantitative results obtained by our approach and the comparison with other methods. It is observed that each of the methods have a much higher failure rate in general, especially over the M.T-II and the Gr.L.R terrains. Our approach significantly outperforms LfD and MfD in terms of failure rate. Similar to the experiments over individual types of terrain, we observe that both LfD methods have slower traversal time for successful runs, but they have a significantly higher failure rate. Moreover, LfD and MfD are outperformed with respect to inconsistency and jerkiness, with MfD performing worst among all tested methods, especially on the jerkiness metric.

Table 7.2 Quantitative results for scenarios when the robot traverses over *complex unstructured off-road terrain* shown in Figure 7.4.

	Failure Rate (/10)				Traversal Time (s)				Inconsistency				Jerkiness (m/s^3)			
Terrain	LfD	MLfD	TRAL	**Ours**	LfD	MLfD	TRAL	**Ours**	LfD	MfD	TRAL	**Ours**	LfD	MfD	TRAL	**Ours**
Gr.M.R	5	7	2	**1**	22.0	**19.7**	27.5	23.1	15.62	17.28	14.54	**12.31**	65.01	80.56	58.36	**51.93**
Gr.L.R	8	9	3	**3**	**27.2**	27.4	29.4	28.8	93.53	101.26	68.87	**51.16**	34.96	40.51	28.22	**24.55**
M.T-I	0	1	0	**0**	**17.9**	18.2	19.4	18.9	3.97	5.38	4.91	**3.39**	72.37	83.17	70.36	**68.55**
M.T-II	5	7	**4**	5	23.1	**18.1**	30.2	28.5	93.37	95.47	80.43	**78.82**	54.13	77.49	52.51	**47.93**

Although both TRAL and our method obtain promising performance, as the main goal is to enhance consistent maneuverability, our approach obtains an average of 15.83% less on inconsistency over TRAL (with 7.66% less over traversal time, 4.50% less over failure rate, and

4.56% less on jerkiness). The p-value for inconsistency improvement is 0.004, indicating that the improvement is statistically significant. This improvement is most likely to be caused by the closed loop feedback, as it is the biggest fundamental difference between the methods.

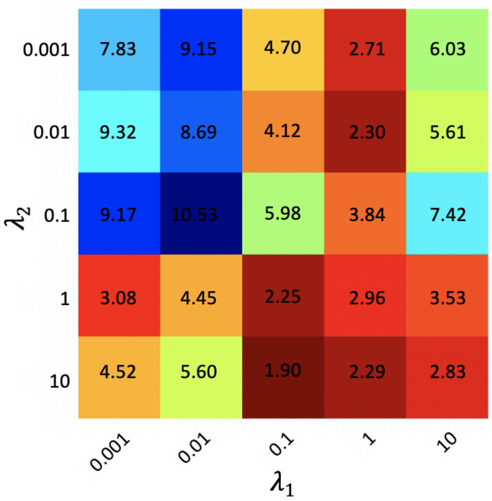


Figure 7.5 Hyperparameter Analysis.

7.6.3 Discussion

Hyperparameter Analysis: The hyperparameters λ_1 and λ_2 in Eq. (7.5) are implemented to balance the loss function and regularization terms. Figure 7.5 depicts how the inconsistency metric changes given varying λ values based on cross-validation during training. It is observed that $\lambda_1 \in (0.1, 10)$ and $\lambda_2 \in (1, 10)$ result in good performance in general. The best result is obtained when $\lambda_1 = 0.1$ and $\lambda_2 = 10$. These values of hyperparameters are then used for all the experiments.

Dependence on Frame Sequences: Our approach uses a sequence of historical frames with length c to generate consistent behaviors. Figure 7.6 shows the change of the inconsistency metric according to c. It is observed that our approach generally performs well when $c \in (15, 20)$, and we observe that the inconsistency metric is worst when either a small number ($c < 5$) or a big number ($c > 30$) is used under the sensing framerate of 30 Hz. These values can be mainly affected by the robot's speed and can differ between robotic platforms.

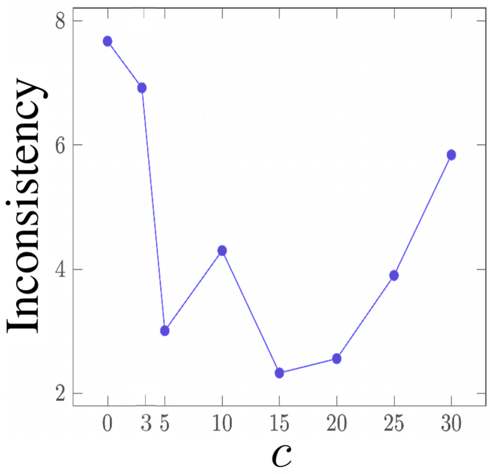


Figure 7.6 Seq. length.

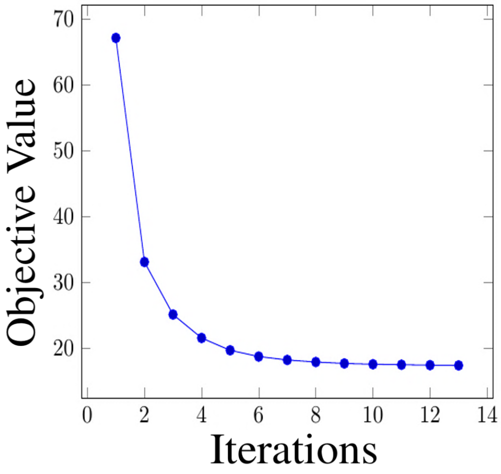


Figure 7.7 Convergence.

Convergence: Experimental results in Figure 7.7 illustrate the fast, monotonically decreasing convergence of Algorithm 7, which validates our theoretical analysis.

Results on Discriminative Feature Modalities: Our approach has the ability to automatically estimate the importance of various feature modalities during the training phase. The visualization of the multimodal features used in our experiments is provided in Figure 7.10. The importance of feature modalities is estimated from the optimal weight tensor $\mathcal{W}$, which is learned on the training data from all individual types of unstructured terrains. As no additional training is performed for the complex off-road unstructured terrains, the importance of feature modalities remains the same for both experimental scenarios.

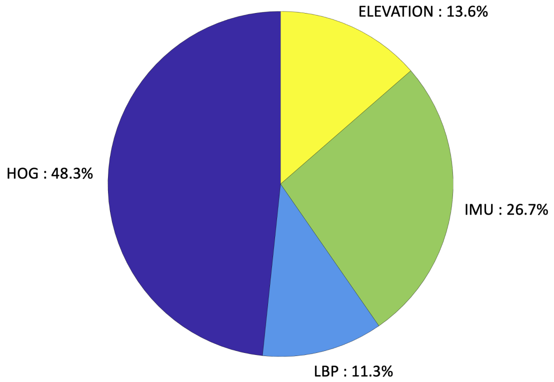


Figure 7.8 Relative importance of the feature modalities estimated by our approach to enable consistent off-road ground navigation.

The results on the importance of feature modalities are demonstrated in Figure 7.8. It is observed that the HOG features are the most important and have a relative importance of 48.3%. The IMU features are observed to be the second most important feature modality with a relative importance of 26.7%. The grid-wise elevation features extracted from a robot's LiDAR sensor are relatively less important and have a relative importance of 13.6%. The least important feature modality is LBP and has a relative importance of 11.3%.

The HOG features have relatively high importance because they capture the terrain shape information, which better characterize the terrain compared to terrain textures encoded by the LBP features. During training, the robot is controlled by a human demonstrator in a manner that reduces the jerkiness of robot navigation. This jerkiness metric is more correlated with the IMU data. Thus, we observe that the IMU features are also of high importance. The results in Figure 7.8 also show that elevation maps are less important. This is probably caused by the fact that the used grid size (25 cm $\times$ 25 cm) cannot capture the essential characteristics of off-road terrains.

7.7 Implementation of the Algorithm and Robot Software System

7.7.1 Algorithm Implementation and Training/Execution Procedures

The optimization algorithm is executed during the offline training phase, and no online optimization is used at the test/execution phase. We provide the terrain feature tensor $\boldsymbol{\mathcal{X}}$, the expected behaviors $\mathbf{Y}$, the actual behaviors $\mathbf{A}$, and the behavior difference tensor $\boldsymbol{\mathcal{E}}$ as input variables to our optimization algorithm. For each expected behavior $\mathbf{Y}$, the robot estimates its actual behavior $\mathbf{A}$, so the dimensionality of $\mathbf{A}$ and $\mathbf{Y}$ is the same. Our algorithm then alternatively optimizes weight tensors $\boldsymbol{\mathcal{W}}$ and $\boldsymbol{\mathcal{U}}$, over various iterations and converges to an optimal solution.

We train our approach on a dataset recorded while an expert (a human in our case) demonstrates robot driving over individual types of terrains, including grass, sand, gravel, medium-sized rocks, and large-sized rocks. The recorded data includes the robot's observations from IMU, RGBD camera, and LiDAR sensors, the robot's actual behavior, and the expected behavior demonstrated by the expert. The training dataset includes approximately 20,000 instances from nearly 5.5 hours of driving. Each instance includes the inputs from c time steps as defined in the main paper. This dataset is used to train our algorithm and identify the optimal hyperparameter values (Figs. 5 and 6 in the main paper) in the training phase. The optimal hyper-parameter we obtain for our training data are: $\lambda_1 = 0.1$, $\lambda_2 = 0.1$ and $c = 15$. In the testing phase, our approach uses the parameters learned during the training phase without additional online learning/optimization. Although our approach is trained using data obtained from individual types of terrains, we include unseen off-road terrains (e.g., grass-medium rocks, and mixed terrains) during testing to evaluate our approach's ability to let ground robots generate consistent behaviors in unfamiliar terrains.

The computational cost increases quadratically with c in each iteration of our algorithm. The number of iterations to convergence may also increase with the increase of c (which unfortunately does not have a mathematical bound). In practice, when the algorithm is trained using the dataset on the robot's onboard computer that has a 4.3 GHz Intel i7 processor and 16GB memory with no GPU, it takes nearly 3.5 hours for the algorithm to converge when $c = 15$, and nearly 21 hours when $c = 40$. We did not test the cases when $c > 40$ because the performance has already significantly dropped when $c > 30$ (Figure 6). The computational cost is linearly dependent on n.

Similar to most ML methods, more training data from more diverse scenarios (e.g., diverse terrains for ground navigation in our case) often improves the results. However, training with more data also increases computational cost. It is not easy to determine the impacts of data quality on the results unfortunately. First, it is difficult to estimate data quality as it is affected by many factors such as sensor types and quality (e.g., resolution and noise), feature extraction, and human/AI-agent demos. Second, the learning model is optimized based on the current quality of data. When data quality changes (e.g., by switching to better feature extractions), the model may not be applicable anymore.

7.7.2 Robot Software System Implementation

We use a Clearpath Husky robot in our field experiments. The robot runs on Ubuntu 18.04 and Robot Operating System (ROS) Melodic as the operating system. Our approach is implemented using C++ as a ROS package, which runs in real time at 30 Hz on an Intel 4.3 GHz i7 processor onboard the robot without requring a GPU. The implementation code is provided as a part of the supplementary material. Instructions of executing our ROS package is also included in the submitted implementation code.

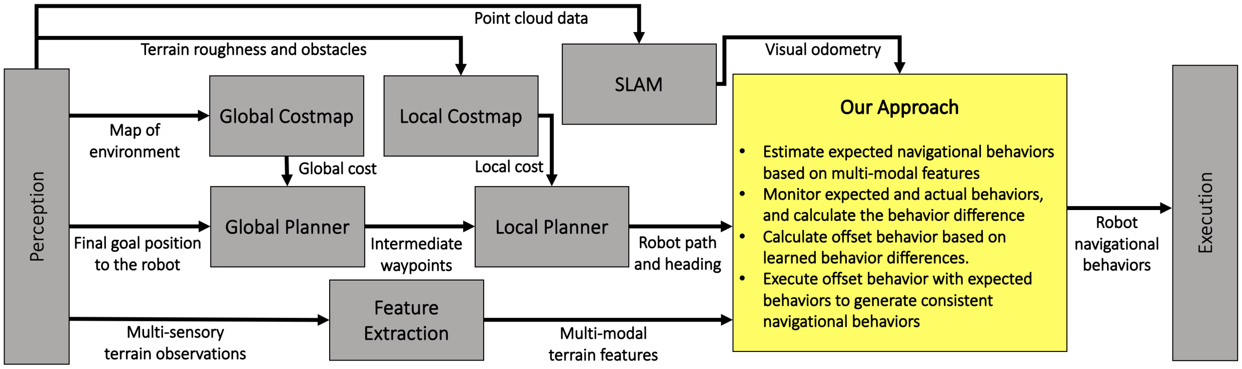

Point cloud data
Terrain roughness and obstacles
SLAM
Visual odometry
Perception
Global Costmap
Local Costmap
Map of environment
Global cost
Local cost
Global Planner
Local Planner
Final goal position to the robot
Intermediate waypoints
Robot path and heading
Feature Extraction
Multi-sensory terrain observations
Multi-modal terrain features
Our Approach
• Estimate expected navigational behaviors based on multi-modal features
• Monitor expected and actual behaviors, and calculate the behavior difference
• Calculate offset behavior based on learned behavior differences.
• Execute offset behavior with expected behaviors to generate consistent navigational behaviors
Robot navigational behaviors
Execution

Figure 7.9 Overview of the ROS-based software architecture implemented on our physical robots for performing consistent behavior generation. The yellow box shows our approach and the remaining modules that work with the proposed approach in our software architecture are shown by gray boxes.

Figure 7.9 provides an overview of the ROS-based software architecture implemented on our physical robots to perform consistent navigational behavior generation. Implemented as a ROS package, our approach works with many other packages within ROS. Our approach works as a local controller to generate navigational controls, and it works together with the packages of global and local planners. The global planner generates a global path using a cost map and offers the local planner intermediate way-points that lead the robot to the goal position. The local planner employs point cloud data from the perception package to detect obstacles near the robot. Then, it generates a local path for the robot to avoid the obstacles between the way-points. The local path calculated from the local planner is then passed onto our approach, which generates consistent navigational behaviors to make the robot to follow the path. Specifically, our approach computes expected navigational behaviors from terrain feature vectors extracted by the feature extraction package. Our approach continuously monitors the difference between the robot's expected behaviors and actual behaviors that are estimated using visual odometry or SLAM. Then, then our approach calculates offset behaviors based on a sequence of past behavior differences. These offset behaviors are applied to compensate for the behavior differences in order to achieve consistent ground navigation when the robot traverse over unstructured off-road terrains.

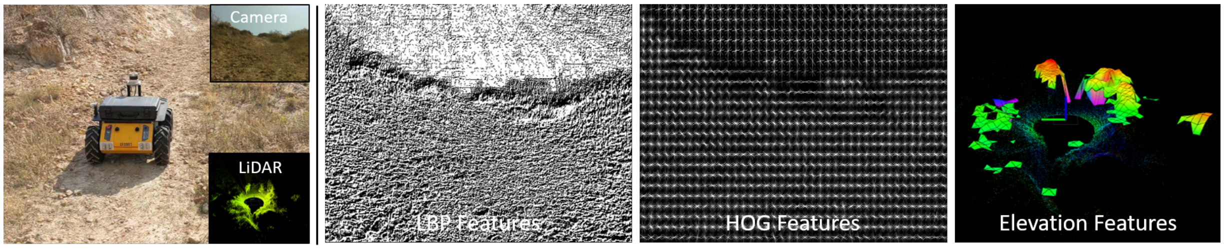


Figure 7.10 Multimodal features used to characterized terrain during ground navigation.

7.8 Assumptions and Limitations

Our approach is based on two main assumptions: First, we assume the LiDAR-based SLAM method used in our autonomy stack as seen in Figure 7.9, provides an accurate estimation of the robot's actual behaviors. However, this assumption may not be satisfied when robots operate in feature-sparse environments such as a hallway with white walls. With inaccurate SLAM, the generated offset behaviors can be sub-optimal. On the other hand, our focus is the unstructured off-road environment (e.g., in a forest), which is often feature rich. So our assumption is satisfied. If not, we may use SLAM methods designed for feature-sparse scenarios [289, 290] or use a Visual Inertial Navigation System (VINS) [291] in environments with visual texture. The second assumption we make is that the dynamics of the robot do not dramatically change over a shorter period of time (e.g., within one second), and the non-linearity in the robot is not severe. This assumption holds as our approach only works as a local controller and moreover, can be easily satisfied when we use robot observations from the past 0.5 seconds, i.e., a sequence length of $c = 15$ frames.

In addition, our approach also makes common assumptions such as the expert demonstrations are reasonably good and robot functions correctly without failures in actuation. Different sensor periodicities are often addressed at the robot's hardware level, e.g., using a MasterClock in our case (which is a precise timing system to synchronize all sensor measurements). In addition, our approach does not explicitly assume that all sensor measurements are always available, but it also does not explicitly address the problem of missing data (e.g., caused by sensor failure or occlusion). Implicitly, our approach is robust to missing data because it fuses measurements from multiple sensors to generate navigational behaviors. That is, if one sensor fails we still have information from the other sensors to generate behaviors. Each sensor contributes differently to navigational behavior generation. For example from Figure 7.8, we observe that features obtained from the color camera (HOG and LBP) have a relative importance of $\sim 60\%$, and features from IMU measurements have a relative importance of $\sim 27\%$. Thus, missing measurements from the color camera is more severe than missing IMU data.

Our approach has two limitations: First, our approach is a local controller that works with an external local planner, as seen in Figure 7.9. This local planner does not have terrain awareness, which caused most failures in our experiments, including the failure case in the video. The second limitation of our approach is from the assumption that the LiDAR-based SLAM method in the robot's autonomy can provide good estimations of the robot's actual behavior. However, as mentioned earlier, this assumption doesn't hold in feature-sparse environments and would lead our approach to generate sub-optimal offset behaviors.

7.9 Conclusion

In this paper, we introduce a novel approach for consistent behavior generation that enables ground robots' actual behavior to more accurately match expected behaviors while adapting to a variety of unstructured off-road terrain. Our approach learns offset behaviors in a self-supervised fashion to compensate for the inconsistency between the actual and expected navigational behaviors without the need to explicitly model various setbacks, and learns the importance of the multi-modal features to improve the representation of terrain for better adaptation. Our proposed approach is extensively evaluated in real-world off-road environments. Experimental results have shown that our approach enables a robot to improve its ground maneuverability when traversing over complex unstructured off-road terrain with more behavior consistency and smoothness compared to previous methods.

CHAPTER 8

SELF-REFLECTIVE TERRAIN-AWARE ROBOT ADAPTATION FOR CONSISTENT OFF-ROAD GROUND NAVIGATION

This work was partially supported by NSF CAREER Award IIS-1942056, NSF CNS-1823245, and the DEVCOM ARL SARA Program W911NF-20-2-0107.

Ground robots require the crucial capability of traversing unstructured and unprepared terrains and avoiding obstacles to complete tasks in real-world robotics applications such as disaster response. When a robot operates in off-road field environments such as forests, the robot's actual behaviors often do not match its expected or planned behaviors, due to changes in the characteristics of terrains and the robot itself. Therefore, the capability of robot adaptation for consistent behavior generation is essential for maneuverability on unstructured off-road terrains. In order to address the challenge, we propose a novel method of self-reflective terrain-aware adaptation for ground robots to generate consistent controls to navigate over unstructured off-road terrains, which enables robots to more accurately execute the expected behaviors through robot self-reflection while adapting to varying unstructured terrains. To evaluate our method's performance, we conduct extensive experiments using real ground robots with various functionality changes over diverse unstructured off-road terrains. The comprehensive experimental results have shown that our self-reflective terrain-aware adaptation method enables ground robots to generate consistent navigational behaviors and outperforms the compared previous and baseline techniques.

8.1 Introduction

Autonomous ground robots require the crucial capability of navigating over unstructured and unprepared terrains in off-road environments and avoiding obstacles to complete tasks in real-world applications such as search and rescue, disaster response, and reconnaissance [107, 292–294]. When operating in field environments, ground robots need to navigate over a wide variety of unstructured off-road terrains with changing types, slope, friction, and other characteristics that cannot be fully modeled beforehand. In addition, ground robots operating over a long-period of time often experience changes in their own functionalities, including damages (e.g., failed robot joints and flat tires), natural wear and tear (e.g., reduced tire traction), and varying robot configurations (e.g., varying payload). As illustrated in Figure 8.1, such challenges make robot navigation in unstructured off-road environments a challenging problem. Thus, robot adaptation to changes in terrains and robot functionalities is essential to the success of off-road ground navigation.

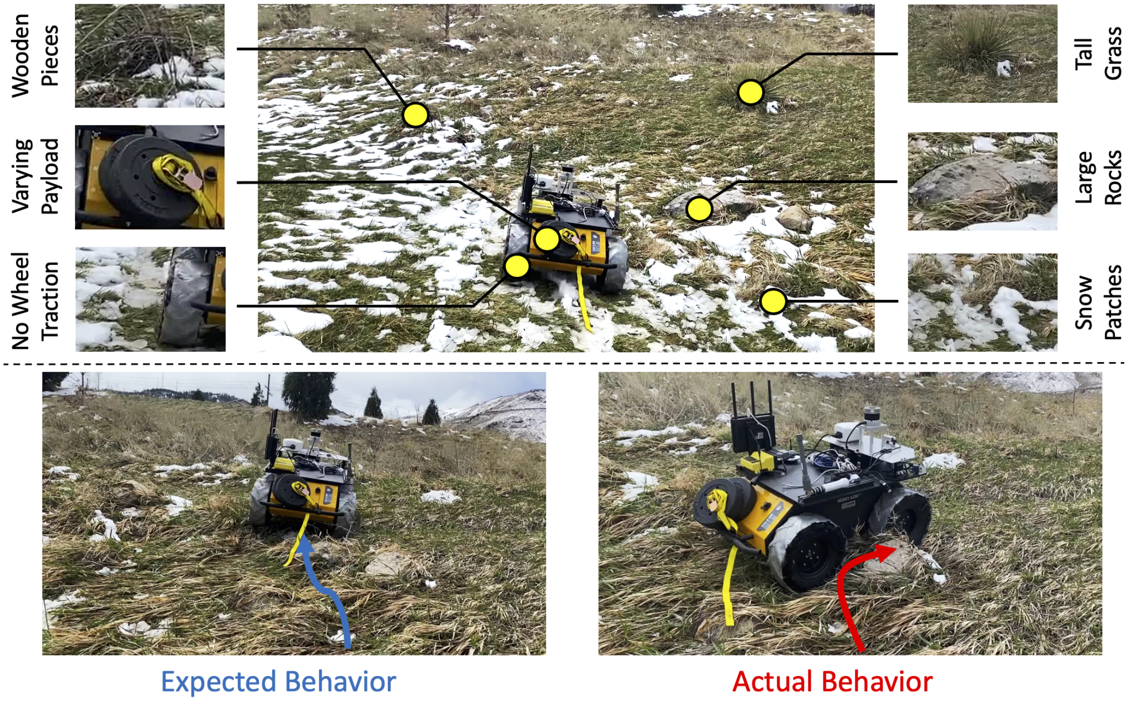


Figure 8.1 A motivating scenario of self-reflective terrain-aware robot adaptation for generating consistent robot navigational behaviors. When robots operate in unstructured off-road environments, the actual behaviors often do not match their expected behaviors, due to changes in the characteristics of terrains and the robots themselves. Therefore, the capability of robot adaptation for consistent behavior generation is essential for maneuverability over unstructured off-road terrains.

Given its importance, robot adaptation to unstructured off-road terrains has been an active area of research over the past decades. Previous methods can be broadly categorized into model-based and learning-based approaches. Model-based methods often use control theories to model a ground robot's dynamics for navigation [295, 296]. However, they typically ignore terrain characteristics.

Learning-based approaches can learn terrain complexity and estimate the possibility of traversing terrains through terrain recognition [118, 267, 268]. Learning-based methods were also developed to adapt navigational behaviors according to the environment [36, 297], and to transfer expert demonstrations on navigational controls to autonomous ground robots based upon learning from demonstration (LfD) [6, 109, 117].

Although previous learning-based approaches have shown promising performance, the expected navigational behaviors generated by previous approaches cannot always be executed accurately by a robot when it navigates over unstructured off-road terrains. That is, a robot's actual navigational behaviors are typically not consistent with the expected controls. This inconsistency is caused primarily by two reasons. First, while ground robots navigate in real-world off-road environments, they experience terrains with a wide variety of characteristics that cannot be modeled beforehand, e.g., tall grass terrain with hidden rocks as illustrated in Figure 8.1. Second, robots may experience negative effects or so-called setbacks [270, 271], which are defined as negative changes in a robot's functionalities that increase the difficulty for the robot to achieve its expected behaviors. Examples of possible setbacks include reduced wheel traction and heavy payload. The previous approaches were generally not able to learn to simultaneously adapt to changes in both terrains and setbacks.

To address these difficulties, we develop a novel method of self-reflective terrain-aware robot adaptation for consistent navigational behavior generation, which enables robots to accurately execute the expected behaviors through robot self-reflection while adapting to varying unstructured off-road terrains. In psychology, self-reflection is considered as the humans' ability to modulate our behaviors by being aware of ourselves [298–300]. We use the term self-reflection in robotics to refer to the robot's capability of adopting self-awareness (e.g., the robot knows it has a slower speed than expected) to modulate navigational behavior generation. To enable robot self-reflection, our method monitors the difference between the expected and actual behaviors, and then adapts the robot's navigational behaviors accordingly to minimize this difference for consistent navigation. Moreover, our method learns representations of unstructured terrains from the robot's multi-sensory observations, which are used to recognize unstructured terrains and generate navigational behaviors.

Our method also fuses the historical observations and robot behaviors that affect the robot's current behavior generation. All the above components are integrated into a unified mathematical framework of constrained regularized optimization with a theoretical convergence guarantee.

The contribution of this paper focuses on the introduction of the first method of self-reflective terrain-aware adaptation for ground robots to generate consistent navigational controls to traverse unstructured off-road environments. The specific novelties of this paper include:

- We introduce a novel method for terrain-aware robot adaptation to unstructured off-road terrains, which is able to simultaneously recognize discriminative terrain features and adaptively generate corresponding ground navigation behaviors.

- We introduce a novel idea of robot self-reflection and implement one of the first learning-based approaches that enable the ground robot to learn an offset control to compensate for the behavior differences in order to enhance consistent ground maneuverability.

- We implement an optimization algorithm to effectively address the constrained regularized optimization problem that is formulated to enable self-reflective terrain-aware robot adaptation, which holds a theoretical guarantee to converge to the global optimal solution.

Furthermore, as an experimental contribution, we perform extensive experiments and provide a comprehensive performance evaluation of terrain adaptation methods. We design a set of scenarios for ground robots with various functionality changes to traverse diverse individual and complex unstructured off-road terrains in natural field environments.

The remainder of this paper is structured as follows. We provide a review of related work in Section 8.2. Our proposed methods for self-reflective terrain-aware robot adaptation is discussed in Section 8.3. We derive the optimization algorithm in Section 8.4. After discussing experimental results in Section 8.5, we provide concluding thoughts in Section 8.6.

8.2 Related Work

In this section, we provide a review of related work in the field of robot adaptation to unstructured terrains, including control and learning-based methods. We also offer a review of recent work related to robot self-reflection.

8.2.1 Control Methods for Ground Navigation

Methods based on classical control theories typically use pre-defined models to generate ground navigation behaviors in order to reach a desired goal position. Many early methods used a fuzzy logic-based implementation [226, 227] without considering prior knowledge of robot dynamics to perform terrain navigation. System identification models were developed that use robot trajectory data to learn the robot's dynamics and accordingly perform ground navigation [228–230]. Trajectory optimization was also designed for terrain navigation through differential dynamics programming (DDP) [231, 232] or iterative linear quadratic regulators (iLQR) [233, 234]. Both learn non-linear robot dynamics models from the robot's interactions with terrains. Closed loop feedback control, such as Model Predictive Control (MPC) [235, 236] and Model Predictive Path Integral (MPPI) [161], were also designed to improve control robustness to terrain noise and robot model error.

These classic control approaches model the robot only and usually lack an awareness of the environment, i.e., they are not terrain-aware. Without terrain awareness, ground robots cannot effectively adapt to unstructured terrains in natural field environments.

8.2.2 Learning-Based Methods for Navigation

Learning-based techniques for navigational behavior generation and terrain adaptation have gained significant attention over the past years because of their effectiveness and flexibility [277].

Pure Data-Driven Learning. Methods based on pure data-driven learning trains a machine learning model from the robot's observations or past experience only to generate navigational behaviors. Methods were designed to learn a robot behavior model using high-dimensional observations [187, 188]. Learning from demonstration (LfD) techniques [80, 237] were widely studied to transfer expert knowledge to mobile robots for ground navigation [117, 192]. Terrain-aware navigation was also implemented by combining representation learning for terrain recognition along with apprenticeship learning to perform terrain adaptation [6]. Navigational affordances were learned from experts over different terrains for navigation [164]. External disturbances were also considered to enable robust ground navigation in unstructured environments [280–282]. Reinforcement learning is applied to enable robot ground navigation, which learns from the robot's own experience in a trial-and-error fashion when navigating in the environment [165, 166, 191]. Rapid terrain-aware adaptation can be achieved by updating learned policies via inferring key terrain parameters [167]. Lifelong reinforcement learning can improve the ground navigation performance by continuously optimizing the learned models [170, 192, 238].

Machine-Learning-Based Control.

Methods based on machine-learning-based control assume a control model and learn the parameter values of the control model from the robot's observations or past experience [194, 195, 239, 240]. Several approaches were implemented based upon Dynamics Mode Decomposition (DMD) [196, 197] and Sparse Identification of Non-Linear Dynamics (SINDy) [198, 241] to learn control models for system identification and ground navigation [199, 200]. Evolutionary algorithms were applied to estimate parameter values of the robot's control model in an online fashion for ground navigation [201, 202]. For robots with a high degree of freedom, methods were implemented to integrate the iterative Linear Quadratic Regulators (iLQR) with machine learning to explore control model parameters for adaptive navigation [203, 242]. Similarly, learning-based control methods were also designed that use a neural network as a functional approximator to represent the robot's dynamics model whose parameters can be updated in an online fashion [204].

Although learning-based methods have shown promising performance for ground navigation in unstructured environments, they usually only adapt to external environments, and lack the key ability to reason about the changes in the robot itself. Accordingly, they are not able to address the inconsistency caused by any robot setbacks, such as reduced traction or increased payload.

8.2.3 Self-Reflective Robot Adaptation

Self-reflective adaptation methods enable a robot to be aware of its own functionalities and capabilities to adapt the robot's behaviors [222, 223, 301]. Early self-reflection methods were used for robots to perform self-supervised improvements while operating in outdoor environments [302]. Methods were also developed to enable a robot to reflect on the past navigational experience to prioritize a set of policies in order to improve robot navigation [303]. To adaptively respond to changes in the environment, a self-reflective risk-aware artificial cognitive model was developed [304]. Payload changes were considered in a self-reflection approach to update the robot's dynamics model for adaptive navigation planning [305]. Robot self-reflection was used in [256] to learn navigational behaviors in cluttered environment by reflecting on the perceptual data from a 2D-LiDAR sensors. Recently, a self-reflection method was also designed to iteratively update the kinematics model of a soft-robot based on its interaction with the environment [306]. A deep learning method was implemented to perform reflection on the robot's trajectory from perceptional data in order to plan the robot's path towards the goal [258].

Most existing self-reflective robot adaptation approaches focused on adapting the robot's behaviors only to changes in the robot's functionalities. But they are generally not able to simultaneously adapt to external environments. Although our preliminary work [34] partially addresses this issue, it models terrain and robot functionality changes in a single loss function, and thus cannot distinguish the effects of terrains and robot functionalities on ground navigation. In this paper, we propose a new robot navigation method that explicitly and jointly characterize the terrain for navigation (i.e., terrain-awareness) and adapt to changes in the robot's functionalities (i.e., self-reflection).

8.3 Approach

In this section, we discuss our novel principled approach to enable ground robots to adapt their navigational behaviors to unstructured off-road terrains and perform self-reflection to generate consistent navigational behaviors. An overview of our approach is shown in Figure 8.2.

Notation: We denote scalars using lowercase italic letters (e.g., $m \in \mathbb{R}$), vectors using boldface lowercase letters (e.g., $\mathbf{m} \in \mathbb{R}^p$), matrices using boldface capital letters, e.g., $\mathbf{M} = \{m_j^i\} \in \mathbb{R}^{p \times q}$ with its i-th row and j-th column denoted as $\mathbf{m}^i$ and $\mathbf{m}_j$, respectively. We use boldface capital Euler script letters to denote tensors (i.e., 3D matrices), e.g., $\boldsymbol{\mathcal{M}} = \{m_j^{i(k)}\} \in \mathbb{R}^{p \times q \times r}$. Unstacking tensor $\boldsymbol{\mathcal{M}}$ along its height (p), width (q) and depth (r) provides slices of matrices $\mathbf{M}^i$, $\mathbf{M}_j$ and $\mathbf{M}^{(k)}$, respectively [307]. The key variables used in our formulation are defined and briefly explained in Table 8.1.

Table 8.1 Definition of the key variables used in our approach.

Variables	Definition
$\mathbf{x}_t \in \mathbb{R}^d$	Terrain feature vector extracted at time t
$\mathbf{X} \in \mathbb{R}^{d \times c}$	Terrain feature instance matrix
$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d \times n \times c}$	Feature tensor from n-different instances
$\mathbf{Z} \in \mathbb{Z}^{l \times n}$	Terrain indicator matrix corresponding to $\boldsymbol{\mathcal{X}}$
$\mathbf{Y} \in \mathbb{R}^{b \times n}$	Expected robot navigational behaviors for $\boldsymbol{\mathcal{X}}$
$\mathbf{A} \in \mathbb{R}^{b \times n}$	Actual robot navigational behaviors for $\boldsymbol{\mathcal{X}}$
$\boldsymbol{\mathcal{E}} \in \mathbb{R}^{b \times n \times c}$	Behavior difference tensor
$\boldsymbol{\mathcal{W}} \in \mathbb{R}^{l \times d \times c}$	Weight tensor to encode terrain features
$\boldsymbol{\mathcal{V}} \in \mathbb{R}^{b \times d \times c}$	Weight tensor to learn estimated behaviors
$\boldsymbol{\mathcal{U}} \in \mathbb{R}^{b \times b \times c}$	Weight tensor to learn offset behaviors
d	Dimensionality of the feature vector
c	Past time steps used to acquire observations
n	Number of training instances
l	Number of terrain types
b	Number of robot behavior controls

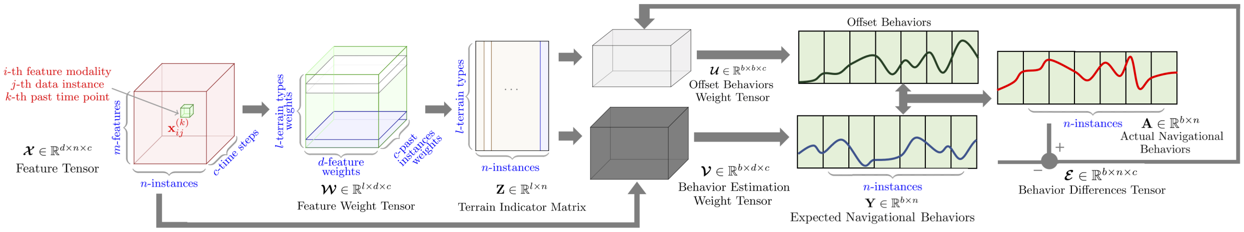

i-th feature modality
j-th data instance
k-th past time point
$\mathbf{x}_{ij}^{(k)}$
m-features
$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d \times n \times c}$
Feature Tensor
n-instances
c-time steps
l-terrain types weights
d-feature weights
c-past instances weights
$\boldsymbol{\mathcal{W}} \in \mathbb{R}^{l \times d \times c}$
Feature Weight Tensor
l-terrain types
n-instances
$\mathbf{Z} \in \mathbb{R}^{l \times n}$
Terrain Indicator Matrix
$\boldsymbol{\mathcal{U}} \in \mathbb{R}^{b \times b \times c}$
Offset Behaviors Weight Tensor
$\boldsymbol{\mathcal{V}} \in \mathbb{R}^{b \times d \times c}$
Behavior Estimation Weight Tensor
Offset Behaviors
n-instances
$\mathbf{Y} \in \mathbb{R}^{b \times n}$
Expected Navigational Behaviors
n-instances
$\mathbf{A} \in \mathbb{R}^{b \times n}$
Actual Navigational Behaviors
+
−
$\boldsymbol{\mathcal{E}} \in \mathbb{R}^{b \times n \times c}$
Behavior Differences Tensor

Figure 8.2 Overview of our proposed method for self-reflective terrain-aware robot adaptation to enable ground robots to adapt their navigational behaviors to unstructured off-road terrains, and perform self-reflection to generate consistent navigational behaviors in the unified constrained regularized optimization framework.

8.3.1 Terrain-Aware Ground Navigation

We use multiple sensors installed on a mobile ground robot to collect observations when the robot traverses unstructured terrains, including RGBD images, LiDAR scans and IMU readings. At each time step t, we extract m types of features (such as visual features and local elevations) from the multi-sensory data. These features are concatenated into a feature vector denoted as $\mathbf{x}^{(t)} \in \mathbb{R}^d$, where $d = \sum_{i=1}^{m} d_i$ with d_i denoting the dimensionality of the i-th feature type. Features extracted from a sequence of c past time steps are stacked into a matrix as a terrain feature instance denoted as $\mathbf{X} = [\mathbf{x}^{(t)}; \ldots; \mathbf{x}^{(t-c)}] \in \mathbb{R}^{d \times c}$. To train our approach, we collect a set of n feature instances that are obtained when the robot traverses over various terrains, and denote this training set as a terrain feature tensor $\boldsymbol{\mathcal{X}} = [\mathbf{X}_1, \ldots, \mathbf{X}_n] \in \mathbb{R}^{d \times n \times c}$.

We denote the terrain types that are associated with $\boldsymbol{\mathcal{X}}$ as $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n] \in \mathbb{Z}^{l \times n}$, where $\mathbf{z}_i \in \mathbb{Z}^l$ denotes the vector of terrain types and l represents the number of terrain types. Each element $z_j^i \in \{0, 1\}$ indicates whether $\mathbf{X}_i$ has the j-th terrain type. During training, ground truth of the terrain types is provided by human experts. Given $\boldsymbol{\mathcal{X}}$ and $\mathbf{Z}$, terrain recognition can be formulated as an optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\mathcal{W}}} \quad & \mathcal{L}(\boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{X}} - \mathbf{Z}) \\ \text{s.t.} \quad & \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{W}}^\top = \boldsymbol{\mathcal{I}} \\ & \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{W}}^\top \otimes (\boldsymbol{\mathcal{W}}^\top)^\top \end{aligned} \tag{8.1}$$

where $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{l \times d \times c}$ is a weight tensor used to encode the importance of each element in $\boldsymbol{\mathcal{X}}$ towards estimating terrain types. Each tensor element $w_j^{i(k)} \in \boldsymbol{\mathcal{W}}$ denotes the weight of the j-th terrain feature from the k-th past time step to recognize the i-th terrain type. $\mathbf{W}^i$, $\mathbf{W}_j$ and $\mathbf{W}^{(k)}$ are the slices of matrices that are obtained by unstacking $\boldsymbol{\mathcal{W}}$ along its height, width and depth, respectively. $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{l \times d \times c}$ is the identity tensor [308], which is used to mathematically define the orthogonality in tensors. The operator $\otimes$ denotes the vector Kronecker tensor product that performs the matrix-wise multiplication of the two tensors [309]. In Eq. (8.1), the tensor product $\otimes$ takes each terrain feature instance $\mathbf{X}_i \in \boldsymbol{\mathcal{X}}$, and multiplies it with the weight tensor $\boldsymbol{\mathcal{W}}$.

The Log-Cosh loss, i.e., $\mathcal{L}(\cdot) = log(cosh(\cdot))$ [310, 311], is used in the objective function in Eq. (8.1) to encode the error of using terrain features $\boldsymbol{\mathcal{X}}$ to recognize terrain types $\mathbf{Z}$, through the learning model that is parameterized by $\boldsymbol{\mathcal{W}}$. The advantage of using the Log-Cosh loss (e.g., over the ℓ_2-loss) is twofold. First, the Log-Cosh loss is robust to outliers, because a linear error becomes a much smaller value using the logarithmic scale, thus it can reduce the learning bias over outliers. Because observations acquired by a robot when navigating over unstructured off-road terrain are usually noisy, the use of the Log-Cosh loss is desirable. Second, $\mathcal{L}(\cdot)$ provides a better optimization of $\boldsymbol{\mathcal{W}}$ near the optimality. This is because, near the optimal value of $\boldsymbol{\mathcal{W}}$, the objective value computed by $\mathcal{L}(\cdot)$ in Eq. (8.1) can change significantly when $\boldsymbol{\mathcal{W}}$ shows small changes [310]. The two constraints together in Eq. (8.1) are the necessary conditions for orthogonality. Orthogonality in tensors makes each slice of matrices that stack up a tensor to be full rank, which means that each slice of matrices is independent and orthogonal (or perpendicular) to each other and thus sharing the least similarity [312]. In our formulation, the orthogonality makes $\mathbf{W}^i, i = 1, \ldots, l$ that correspond to all the terrain types to be independent of each other.

The first main novelty of this paper is that we propose a new method for terrain-aware navigation adaptation by joint terrain classification and behavior learning under a unified constrained optimization framework. This is achieved by projecting an observation's feature space into a terrain type space and then further projecting the terrain type space into a robot navigational behavior space.

Formally, we denote the robot's navigational behaviors as $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n] \in \mathbb{R}^{b \times n}$, where $\mathbf{y}_i \in \mathbb{R}^b$ represents the navigational behaviors that the robot is expected to execute while traversing over terrain instance $\mathbf{X}_i$, and b denotes the number of distinctive robot controls (e.g., linear and angular velocities). The expected behaviors can be obtained through demonstrations recorded from human experts as the experts control (i.e., teleoperate) ground robots to traverse various terrains. Similar to terrain classification, navigational behaviors are learned using a history of observations from the past c time steps, which allows our formulation to implicitly consider the dynamics of a ground robot. Then, the problem of terrain-aware navigation adaptation through joint terrain recognition and behavior generation can be formulated as:

$$\begin{aligned} \min_{\boldsymbol{\mathcal{W}},\boldsymbol{\mathcal{V}}} \quad & \mathcal{L}(\boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{X}} - \mathbf{Z}) + \mathcal{L}(\boldsymbol{\mathcal{V}} \otimes \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{X}} - \mathbf{Y}) \\ \text{s.t.} \quad & \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{W}}^\top = \boldsymbol{\mathcal{I}} \\ & \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{W}}^\top \otimes (\boldsymbol{\mathcal{W}}^\top)^\top \end{aligned} \tag{8.2}$$

where $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{b \times d \times c}$ is a weight tensor that indicates the importance of the input features $\boldsymbol{\mathcal{X}}$ to generate navigational behaviors $\mathbf{Y}$, and each element $v_j^{i(k)}$ represents the weight of using j-th terrain feature from k-th past time step towards generating the i-th navigational behavior. We further adopt $\mathbf{V}^i$, $\mathbf{V}_j$ and $\mathbf{V}^{(k)}$ to denote the slices of matrices obtained by unstacking $\boldsymbol{\mathcal{V}}$ along its height, width and depth, respectively.

The second term in the objective function in Eq. (8.2) is a loss that encodes the difference between the learning model and the demonstrated robot's navigational behaviors. This loss aims to learn a projection from a history of observations to terrain types, and then use the terrain types to generate the robot's behaviors, thus, enabling terrain awareness for our navigational behavior generation method.

Different terrain features capture different characteristics of unstructured terrains, e.g., color, texture, and slope. These features typically have different contributions towards terrain recognition and behavior generation. Similarly, features extracted in specific time steps can be more important than others. Thus, it is essential to identify the most discriminative features when multiple types of features from a history of time steps are used. To achieve this capability, we design a new regularization term called the behavior norm, which is mathematically defined as:

$$\|\boldsymbol{\mathcal{V}} \otimes \boldsymbol{\mathcal{W}}\|_B = \sum_{j=1}^{m} \sum_{k=1}^{c} \|\mathbf{V}_j^{(k)} \mathbf{W}_j^{(k)}\|_F \tag{8.3}$$

where $\mathbf{V}_i^{(k)}$ and $\mathbf{W}_i^{(k)}$ denote the slices of matrices of $\boldsymbol{\mathcal{V}}$ and $\boldsymbol{\mathcal{W}}$ obtained at the k-th past time step for the i-th type of features, respectively. This norm enforces sparsity between feature modalities obtained over the past c time steps in order to identify the most discriminative features and time steps during training.

Finally, the problem of terrain-aware behavior generation for terrain adaptation is formulated as:

$$\begin{aligned}
\min_{\boldsymbol{\mathcal{W}},\boldsymbol{\mathcal{V}}} \quad & \mathcal{L}(\boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{X}} - \mathbf{Z}) + \mathcal{L}(\boldsymbol{\mathcal{V}} \otimes \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{X}} - \mathbf{Y}) \\
& + \lambda_1 \|\boldsymbol{\mathcal{V}} \otimes \boldsymbol{\mathcal{W}}\|_B \\
\text{s.t.} \quad & \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{W}}^\top = \boldsymbol{\mathcal{I}} \\
& \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{W}}^\top \otimes (\boldsymbol{\mathcal{W}}^\top)^\top
\end{aligned} \tag{8.4}$$

where λ_1 is a trade-off hyperparamter to control the amount of regularization. The problem formulation in Eq. (8.4) allows for terrain-aware navigational behavior generation. However, ground robots operating in an unstructured environment over a long-period of time often experience changes in their own functionalities, such as damages (e.g., failed motors and flat tires), natural wear and tear (e.g., reduced tire traction), and changed robot configurations (e.g., varying payload). These negative effects, also called *setbacks*, reduce the effectiveness of robot navigation and cause the robot's actual navigational behaviors to deviate from the planned or expected behaviors.

8.3.2 Self-Reflection for Consistent Navigational Behavior Generation

The second main novelty of this paper is to propose the idea of self-reflection for a ground robot to generate consistent navigational behaviors by adapting to changes in the robot's functionalities. In psychology, self-reflection is considered as the humans' capability of modulating our behaviors by being aware of ourselves [298–300]. We use the term self-reflection in robotics to refer to the robot's capability of adopting self-awareness (e.g., knowing it has slower speeds than expected) to modulate the terrain-aware behavior generation. Through self-reflection, we enable a ground robot to generate terrain-aware navigational behaviors that are also conditioned on its self-awareness. The robot can be self-aware in the way that it continuously monitors the difference between its actual and expected navigational behaviors, where this difference is caused by setbacks or negative effects. Then, our method adapts the parameters of terrain-aware behavior generation according to the robot's self-awareness, and computes an offset control to generate consistent navigational behaviors without explicitly modeling the robot setbacks.

Mathematically, we denote the robot's actual navigational behaviors as $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n] \in \mathbb{R}^{b \times n}$, where $\mathbf{a}_i \in \mathbb{R}^b$ is the actual navigational behaviors corresponding to the i-th instance $\mathbf{X}_i$. The actual behaviors can be estimated when the ground robot navigates over unstructured terrains using pose estimation methods such as SLAM or visual odometry [287, 313]. Due of setbacks, the robot's actual behaviors usually do not match its expected or planned navigational behaviors. We compute the difference in navigational behaviors over the past c time steps as $\mathbf{E} = [(\mathbf{a}^{(t)} - \mathbf{y}^{(t)}), \dots, (\mathbf{a}^{(t-c)} - \mathbf{y}^{(t-c)})] \in \mathbb{R}^{b \times c}$, and further denote the behavior differences for all instances $\boldsymbol{\mathcal{X}}$ as a tensor $\boldsymbol{\mathcal{E}} = [\mathbf{E}_1, \dots, \mathbf{E}_n] \in \mathbb{R}^{b \times n \times c}$.

Then, we introduce a novel loss function to encode self-reflection for achieving consistent terrain-aware navigational behavior generation as follows:

$$\mathcal{L}(\boldsymbol{\mathcal{U}} \otimes \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{E}} - (\mathbf{A} - \mathbf{Y})) \tag{8.5}$$

where $\boldsymbol{\mathcal{U}} = [\mathbf{U}^{(1)}, \dots, \mathbf{U}^{(c)}] \in \mathbb{R}^{b \times b \times c}$ is the weight tensor with $\mathbf{U}^{(k)} \in \mathbb{R}^{b \times c}$ indicating the importance of the k-th past behavior difference $\mathbf{a}^{(k)} - \mathbf{y}^{(k)}$ to generate offset behaviors $\boldsymbol{\mathcal{U}} \otimes \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{E}}$. Because the loss function in Eq. (8.5) encodes the error between the behavior differences and the generated offset behaviors, minimizing this loss function is equivalent to minimizing the behavior differences using the generated offset behaviors, or maximizing the consistency of the actual and expected behaviors. By monitoring the differences over the past c time steps to improve the navigational behavior at the current time step, this loss function allows our approach to be self-reflective and act as a closed-loop controller for consistent behavior generation while being terrain aware.

During robot navigation, historical data from the past time steps often contributes differently towards generating offset behaviors because of inertia. For example, a heavier payload increases the inertia of the robot; thus a longer time history needs to be considered for generating offset behaviors. Thus, we introduce an additional regularization term to learn the most informative behavior differences in the past time steps for efficient self-reflection, which is defined as:

$$\|\boldsymbol{\mathcal{U}}\|_R = \sum_{k=1}^{c} \|\mathbf{U}^{(k)}\|_F \tag{8.6}$$

where $\mathbf{U}^{(k)}$ denotes the slice of matrices obtained from $\boldsymbol{\mathcal{U}}$ at the k-th past time step.

Finally, integrating all components together, we formulate the problem of self-reflective terrain-aware robot adaptation for generating consistent robot navigational behaviors as a regularized optimization problem in a unified mathematical framework:

$$\begin{aligned} \min_{\boldsymbol{\mathcal{W}},\boldsymbol{\mathcal{V}},\boldsymbol{\mathcal{U}}} \quad & \mathcal{L}(\boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{X}} - \mathbf{Z}) + \mathcal{L}(\boldsymbol{\mathcal{V}} \otimes \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{X}} - \mathbf{Y}) \\ & + \mathcal{L}(\boldsymbol{\mathcal{U}} \otimes \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{E}} - (\mathbf{A} - \mathbf{Y})) \\ & + \lambda_1 \|\boldsymbol{\mathcal{V}} \otimes \boldsymbol{\mathcal{W}}\|_B + \lambda_2 \|\boldsymbol{\mathcal{U}}\|_R \\ \text{s.t.} \quad & \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{W}}^\top = \boldsymbol{\mathcal{I}} \\ & \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{W}}^\top \otimes (\boldsymbol{\mathcal{W}}^\top)^\top \end{aligned} \tag{8.7}$$

where λ_2, similar to λ_1, is a hyperparameter to control the trade-off between the loss functions and regularization terms. An illustration of our complete approach for self-reflective terrain-aware robot adaptation is presented in Figure 8.2.

After computing the optimal values of the weight tensors $\boldsymbol{\mathcal{W}}$, $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{V}}$ according to Algorithm 1 in the training phase, a robot can apply our self-reflective terrain-aware adaptation method to generate consistent navigational behaviors during execution. At each time step t in the execution phase, the robot computes multi-model features $\mathbf{X}_t$ from observations obtained by its onboard sensors over the past c time steps. The robot also estimates the corresponding actual behaviors using pose estimation techniques (such as SLAM or visual odometry [287, 313]) and computes the matrix of the behavior differences $\mathbf{E}_t$. Then our approach can be used by the robot to generate self-reflective terrain-aware navigational behaviors by:

$$\mathbf{y} = \boldsymbol{\mathcal{V}} \otimes \boldsymbol{\mathcal{W}} \otimes \mathbf{X}_t + \boldsymbol{\mathcal{U}} \otimes \boldsymbol{\mathcal{W}} \otimes \mathbf{E}_t \tag{8.8}$$

The first term in Eq. (8.8) generates the navigational behaviors that are aware of terrain types, which allows a ground robot to adapt its navigational behaviors to unstructured terrains. The second term in Eq. (8.8) provides offset controls based on the monitoring of the differences between the expected and actual behaviors in order to compensate for the setbacks and improve consistency in

Algorithm 7: The implemented algorithm to solve the formulated constrained regularized optimization problem in Eq. (8.7).

```
Input  : $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d\times n\times c}$, $\mathbf{Y} \in \mathbb{R}^{b\times n}$, $\mathbf{A} \in \mathbb{R}^{b\times n}$, and $\boldsymbol{\mathcal{E}} \in \mathbb{R}^{b\times n\times c}$
Output : The weight tensors $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{l\times d\times c}$, $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{b\times d\times c}$ and $\boldsymbol{\mathcal{U}} \in \mathbb{R}^{b\times b\times c}$
Initialize $\boldsymbol{\mathcal{W}}$, $\boldsymbol{\mathcal{V}}$ and $\boldsymbol{\mathcal{U}}$;
while not converge do
    Calculate the block diagonal matrix $\mathbf{Q}^B$ with $j$-th column and $k$-th row element given as $\frac{\mathbf{I}_1}{\sum_{i=j}^{m}\sum_{k=1}^{c}\|\mathbf{V}_j^{(k)}\mathbf{W}_j^{(k)}\|_F}$;
    Calculate the block diagonal matrix $\mathbf{Q}^l$ with $i$-th diagonal block given as $\frac{\mathbf{I}^l}{\|\mathbf{W}^i(\mathbf{W}^i)^\top-\mathbf{I}^l\|_F}$;
    Calculate the block diagonal matrix $\mathbf{Q}_d$ with $j$-th diagonal block given as $\frac{\mathbf{I}_d}{\|\mathbf{W}_j(\mathbf{W}_j)^\top-\mathbf{I}_d\|_F}$;
    Calculate the block diagonal matrix $\mathbf{Q}^{(c)}$ with $k$-th diagonal block given as $\frac{\mathbf{I}^{(c)}}{\|\mathbf{W}^{(k)}(\mathbf{W}^{(k)})^\top-\mathbf{I}^c\|_F}$;
    Compute each slice of matrices $\mathbf{W}^i$ from Eq. (8.11);
    Calculate the block diagonal matrix $\mathbf{O}^B$ with $j$-th column and $k$-th row element given as $\frac{\mathbf{I}_1}{\sum_{i=j}^{m}\sum_{k=1}^{c}\|\mathbf{V}_j^{(k)}\mathbf{W}_j^{(k)}\|_F}$;
    Compute the matrix $\mathbf{V}^i$ from Eq. (8.12);
    Calculate the block diagonal matrix $\mathbf{P}^{(R)}$ with $k$-th diagonal block given as $\frac{\mathbf{I}_2}{\sum_{k=t}^{t-c}\|\mathbf{U}^{(k)}\|_F}$;
    Compute the matrix $\mathbf{U}^i$ from Eq. (8.13);
return: $\boldsymbol{\mathcal{W}}$, $\boldsymbol{\mathcal{V}}$ and $\boldsymbol{\mathcal{U}}$
```

navigational behaviors.

8.4 Optimization Algorithm

In this section, we implement a new iterative algorithm, as shown in Algorithm 7, to compute the optimal solution to the formulated constrained regularized optimization problem in Eq. (8.7). The optimization problem is challenging to solve because of the constraints added to the objective function and the two non-smooth structured regularization terms.

To solve the constrained optimization problem in Eq. (8.7), we first convert the constraints into a matrix form as:

$$
\begin{aligned}
\min_{\boldsymbol{\mathcal{W}},\boldsymbol{\mathcal{V}},\boldsymbol{\mathcal{U}}} & \mathcal{L}(\boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{X}} - \mathbf{Z}) + \mathcal{L}(\boldsymbol{\mathcal{V}} \otimes \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{X}} - \mathbf{Y}) \\
& + \mathcal{L}(\boldsymbol{\mathcal{U}} \otimes \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{E}} - (\mathbf{A} - \mathbf{Y})) + \lambda_1 \|\boldsymbol{\mathcal{V}} \otimes \boldsymbol{\mathcal{W}}\|_B + \lambda_2 \|\boldsymbol{\mathcal{U}}\|_R \\
\text{s.t} \quad & \mathbf{W}^i (\mathbf{W}^i)^\top = \mathbf{I}^l; \forall i = 1, \ldots, l \\
& \mathbf{W}_j (\mathbf{W}_j)^\top = \mathbf{I}_d; \forall j = 1, \ldots, d \\
& \mathbf{W}^{(k)} (\mathbf{W}^{(k)})^\top = \mathbf{I}^{(c)}; \forall k = 1, \ldots, c
\end{aligned} \tag{8.9}
$$

where $\mathbf{I}^l$, $\mathbf{I}_d$ and $\mathbf{I}^{(c)}$ are identity matrices of size l, d, c. In Eq. (8.9), the constraints are written using a matrix form, This form still enforces orthogonality, the same as the tensor form in Eq. (8.7), as the matrix-form constraints make each column in each slice of matrices $\mathbf{W}^i$, $\mathbf{W}_j$ and $\mathbf{W}^{(k)}$ to be full rank. Thus, each slice of matrix in tensor $\boldsymbol{\mathcal{W}}$ is also full rank and $\boldsymbol{\mathcal{W}}$ is orthogonal.

Then we can rewrite the objective function in Eq. (8.9) using Lagrangian multipliers λ^l, λ_d and $\lambda^{(c)}$ as:

$$
\begin{aligned}
\min_{\boldsymbol{\mathcal{W}},\boldsymbol{\mathcal{V}},\boldsymbol{\mathcal{U}}} & \mathcal{L}(\boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{X}} - \mathbf{Z}) + \mathcal{L}(\boldsymbol{\mathcal{V}} \otimes \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{X}} - \mathbf{Y}) + \mathcal{L}(\boldsymbol{\mathcal{U}} \otimes \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{E}} - (\mathbf{A} - \mathbf{Y})) \\
& + \lambda_1 \|\boldsymbol{\mathcal{V}} \otimes \boldsymbol{\mathcal{W}}\|_B + \lambda_2 \|\boldsymbol{\mathcal{U}}\|_R + \Sigma_{i=1}^{l} \lambda^l \|\mathbf{W}^i (\mathbf{W}^i)^\top - \mathbf{I}^l\|_F \\
& + \Sigma_{j=1}^{d} \lambda_d \|\mathbf{W}_j (\mathbf{W}_j)^\top - \mathbf{I}_d\|_F + \Sigma_{k=1}^{c} \lambda^{(c)} \|\mathbf{W}^{(k)} (\mathbf{W}^{(k)})^\top - \mathbf{I}^{(c)}\|_F
\end{aligned} \tag{8.10}
$$

Each of the Lagrangian multipliers is designed to have a high value (i.e., $\lambda^l, \lambda_d, \lambda^{(c)} \gg 0$), such that even a small variation in orthogonality in each slice of matrix from weight tensor $\boldsymbol{\mathcal{W}}$ can result in a high cost. To compute the optimal weight tensor $\boldsymbol{\mathcal{W}}$, we minimize Eq. (8.10) with respect to $\mathbf{W}^i$, $i = 1, \ldots, l$, resulting in:

$$
\begin{aligned}
& \tanh(\boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{X}} - \mathbf{Z})((\mathbf{X}^i)^\top \mathbf{W}^i) \tanh(\boldsymbol{\mathcal{V}} \otimes \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{X}} - \mathbf{Y})((\mathbf{X}^i \mathbf{V}^i)^\top \mathbf{W}^i) \\
& + \tanh(\boldsymbol{\mathcal{U}} \otimes \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{E}} - (\mathbf{A} - \mathbf{Y}))((\mathbf{E}^i \mathbf{V}^i)^\top \mathbf{W}^i) \\
& + (\lambda_1 \mathbf{Q}^B + \Sigma_{i=1}^{l} \lambda^l \mathbf{Q}^l + \Sigma_{j=1}^{d} \lambda_d \mathbf{Q}_d + \Sigma_{k=1}^{c} \lambda^{(c)} \mathbf{Q}^{(c)}) \mathbf{W}^i = 0
\end{aligned} \tag{8.11}
$$

where $\mathbf{Q}^B, \mathbf{Q}^l, \mathbf{Q}_d$ and $\mathbf{Q}^{(c)}$ denote block diagonal matrices that are dependent on the weight tensor $\boldsymbol{\mathcal{W}}$. Mathematically, we express each element in the j-th column and k-th row in $\mathbf{Q}^B$ as $\frac{1}{\sum_{i=j}^{m} \sum_{k=1}^{c} \|\mathbf{V}_j^{(k)} \mathbf{W}_j^{(k)}\|_F}$. Each of the i-th blocks in $\mathbf{Q}^l$ is given by $\frac{\mathbf{I}^i}{\|\mathbf{W}^i (\mathbf{W}^i)^\top - \mathbf{I}^l\|_F}$. The j-th diagonal block in $\mathbf{Q}_d$ is given as $\frac{\mathbf{I}_d}{\|\mathbf{W}_j(\mathbf{W}_j)^\top - \mathbf{I}_d\|_F}$ and each block diagonal element in $\mathbf{Q}^{(c)}$ is given by $\frac{\mathbf{I}^{(c)}}{\|\mathbf{W}^{(k)}(\mathbf{W}^{(k)})^\top - \mathbf{I}^{(c)}\|_F}$. Because each slice of matrix $\mathbf{W}^i$ and the block diagonal matrices $\mathbf{Q}^B$, $\mathbf{Q}^l$, $\mathbf{Q}_d$, $\mathbf{Q}^{(c)}$ are interdependent, we need an iterative algorithm to compute them.

We use the optimal slices of weight matrices $\mathbf{W}^i$ to solve weight tensors $\boldsymbol{\mathcal{V}}$ and $\boldsymbol{\mathcal{U}}$. Accordingly, we differentiate Eq. (8.10) with respect to $\mathbf{V}^i$ and $\mathbf{U}^i$, $i = 1, \ldots, b$, resulting in:

$$\tanh\left(\boldsymbol{\mathcal{V}} \otimes \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{X}} - \mathbf{Y}\right)((\mathbf{W}^i\mathbf{X}^i)^\top \mathbf{V}^i - \mathbf{W}\mathbf{X}^i\mathbf{y}^i) + \lambda_1 \mathbf{O}^B = 0 \tag{8.12}$$

and

$$\tanh\left(\boldsymbol{\mathcal{U}} \otimes \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{E}} - (\mathbf{A} - \mathbf{Y})\right)((\mathbf{W}^i\mathbf{E}^i)^\top \mathbf{U}^i - \mathbf{E}^i) + \lambda_2 \mathbf{P}^{(R)} = 0 \tag{8.13}$$

where $\mathbf{O}^{(B)}$ and $\mathbf{P}^{(R)}$ represent block diagonal matrices that are dependent on $\boldsymbol{\mathcal{V}}$ and $\boldsymbol{\mathcal{U}}$, respectively. Mathematically, we calculate each element in the j-th column and k-th row in $\mathbf{O}^B$ as $\frac{1}{\sum_{i=j}^{m}\sum_{k=1}^{c}\|\mathbf{V}_j^{(k)}\mathbf{W}_j^{(k)}\|_F}$. The k-th block diagonal element in $\mathbf{P}^{(R)}$ is calculated by $\frac{\mathbf{I}_{(k)}}{\sum_{k=1}^{c}\|\mathbf{U}^{(k)}\|_F}$. The optimal values of $\mathbf{V}^i$ and $\mathbf{U}^i$ are employed to update $\mathbf{W}^i$ in the next iteration. Similarly, $\mathbf{U}^i$ and $\mathbf{P}$ are interdependent, and thus an iterative algorithm is needed to update their values.

In the following, we prove that Algorithm 7 decreases the value of the objective function in Eq. (8.7) with each iteration and converges to the global optimal solution. But first, we present a lemma from [151]:

Lemma 6: Given any two matrices $\mathbf{A}$ and $\mathbf{B}$, the following inequality relation holds:

$$\|\mathbf{B}\|_F - \frac{\|\mathbf{B}\|_F^2}{2\|\mathbf{A}\|_F} \leq \|\mathbf{A}\|_F - \frac{\|\mathbf{A}\|_F^2}{2\|\mathbf{A}\|_F} \tag{8.14}$$

Theorem 6: Algorithm 7 converges to the global optimal solution of the constrained regularized optimization problem in Eq. (8.7).

Proof. According to Step 7 in Algorithm 7, the value of $\mathbf{W}^i(s+1)$ is computed from $\boldsymbol{\mathcal{W}}(s)$ in the s-th iteration by:

$$\begin{aligned}\mathbf{W}^i(s+1) = \mathcal{L}(\boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{X}} - \mathbf{Z}) + \lambda_1 Tr(\mathbf{V}^i \mathbf{W}^{i\top})\mathbf{Q}^B(s+1)(\mathbf{V}^i\mathbf{W}^i)\\ +\lambda^l Tr(\mathbf{W}^i\mathbf{W}^{i\top} - \mathbf{I}^{l^\top})\mathbf{Q}^l(s+1)(\mathbf{W}^i\mathbf{W}^{i\top} - \mathbf{I}^l)\\ +\lambda_d Tr(\mathbf{W}_j\mathbf{W}_j^\top - \mathbf{I}_d)^\top\mathbf{Q}_d(s+1)(\mathbf{W}_j\mathbf{W}_j^\top - \mathbf{I}^d)\\ +\lambda^{(c)} Tr(\mathbf{W}^{(k)}\mathbf{W}^{(k)\top} - \mathbf{I}^{(c)\top})\mathbf{Q}^{(c)}(s+1)(\mathbf{W}^{(k)}\mathbf{W}^{(k)\top} - \mathbf{I}^{(c)})\end{aligned} \tag{8.15}$$

where the operator Tr denotes the trace of a matrix. From Step 9 in Algorithm 7, we obtain:

$$\mathbf{V}^i(s+1) = \mathcal{L}(\boldsymbol{\mathcal{V}} \otimes \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{X}} - \mathbf{Y}) + \lambda_1 Tr(\mathbf{V}^i\mathbf{W}^i)^\top\mathbf{O}^B(s+1)(\mathbf{V}^i\mathbf{W}^i) \tag{8.16}$$

From Step 11 in Algorithm 7 we also obtain:

$$\mathbf{U}^i(s+1) = \mathcal{L}(\boldsymbol{\mathcal{U}} \otimes \boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{E}} - (\mathbf{A} - \mathbf{Y})) + \lambda_2 Tr\mathbf{U}^i\mathbf{P}(s+1)\mathbf{U}^i \tag{8.17}$$

Then, we derive that:

$$\begin{aligned}&\boldsymbol{\mathcal{F}}_1(s+1) + \boldsymbol{\mathcal{F}}_2(s+1) + \boldsymbol{\mathcal{F}}_3(s+1) + \lambda_1 Tr(\mathbf{V}^i\mathbf{W}^i)^\top\mathbf{Q}^B(s+1)(\mathbf{V}^i\mathbf{W}^i)\\ &+ \lambda^l Tr(\mathbf{W}^i\mathbf{W}^{i\top} - \mathbf{I}^l)^\top\mathbf{Q}^l(s+1)(\mathbf{W}^i\mathbf{W}^{i\top} - \mathbf{I}^l)\\ &+ \lambda_d Tr(\mathbf{W}_j\mathbf{W}_j^\top - \mathbf{I}_d)^\top\mathbf{Q}_d(s+1)(\mathbf{W}_j\mathbf{W}_j^\top - \mathbf{I}^d)\\ &+ \lambda^{(c)} Tr(\mathbf{W}^{(k)}\mathbf{W}^{(k)\top} - \mathbf{I}^{(c)\top}\mathbf{Q}^{(c)}(s+1)(\mathbf{W}^{(k)}\mathbf{W}^{(k)\top} - \mathbf{I}^{(c)})\\ &+ \lambda_1 Tr(\mathbf{V}^i\mathbf{W}^i)^\top\mathbf{O}^B(s+1)(\mathbf{V}^i\mathbf{W}^i) + \lambda_2 Tr\mathbf{U}^i\mathbf{P}(s+1)\mathbf{U}^i\\ &\leq \boldsymbol{\mathcal{F}}_1(s) + \boldsymbol{\mathcal{F}}_2(s) + \boldsymbol{\mathcal{F}}_3(s) + \lambda_1 Tr(\mathbf{V}^i\mathbf{W}^i)^\top\mathbf{Q}^B(s)(\mathbf{V}^i\mathbf{W}^i)\\ &+ \lambda^l Tr(\mathbf{W}^i\mathbf{W}^{i\top} - \mathbf{I}^l)^\top\mathbf{Q}^l(s)(\mathbf{W}^i\mathbf{W}^{i\top} - \mathbf{I}^l)\\ &+ \lambda_d Tr(\mathbf{W}_j\mathbf{W}_j^\top - \mathbf{I}_d)^\top\mathbf{Q}_d(s)(\mathbf{W}_j\mathbf{W}_j^\top - \mathbf{I}^d)\\ &+ \lambda^{(c)} Tr(\mathbf{W}^{(k)}\mathbf{W}^{(k)\top} - \mathbf{I}^{(c)})^\top\mathbf{Q}^{(c)}(s)(\mathbf{W}^{(k)}\mathbf{W}^{(k)\top} - \mathbf{I}^{(c)})\\ &+ \lambda_1 Tr(\mathbf{V}^i\mathbf{W}^i)^\top\mathbf{O}^B(s)(\mathbf{V}^i\mathbf{W}^i) + \lambda_2 Tr\mathbf{U}^i\mathbf{P}(s)\mathbf{U}^i\end{aligned} \tag{8.18}$$

where:

$$\begin{aligned}\boldsymbol{\mathcal{F}}_1(s) &= \mathcal{L}(\boldsymbol{\mathcal{W}}(s) \otimes \boldsymbol{\mathcal{X}} - \mathbf{Z})\\ \boldsymbol{\mathcal{F}}_2(s) &= \mathcal{L}(\boldsymbol{\mathcal{V}}(s) \otimes \boldsymbol{\mathcal{W}}(s) \otimes \boldsymbol{\mathcal{X}} - \mathbf{Y})\\ \boldsymbol{\mathcal{F}}_3(s) &= \mathcal{L}(\boldsymbol{\mathcal{U}}(s) \otimes \boldsymbol{\mathcal{W}}(s) \otimes \boldsymbol{\mathcal{E}} - (\mathbf{A} - \mathbf{Y}))\end{aligned}$$

After substituting $\mathbf{Q}^B$, $\mathbf{Q}^l$, $\mathbf{Q}_d$, $\mathbf{Q}^{(c)}$, $\mathbf{O}^B$ and $\mathbf{P}^R$ in Eq. (8.18), we obtain:

$$\begin{aligned}
&\boldsymbol{\mathcal{F}}_1(s+1)+\boldsymbol{\mathcal{F}}_2(s+1)+\boldsymbol{\mathcal{F}}_3(s+1)+\lambda_2\sum_{k=1}^{c}\frac{\|\mathbf{U}^{(k)}(s+1)\|_F^2}{2\|\mathbf{U}^{(k)}(s)\|_F}\\
&+\lambda_1\sum_{i=j}^{m}\sum_{k=1}^{c}\frac{\|\mathbf{V}_j^{(k)}(s)\mathbf{W}_j^{(k)}(s+1)\|_F^2}{2\|\mathbf{V}_j^{(k)}(s)\mathbf{W}_j^{(k)}(s)\|_F}\\
&+\lambda^l\sum_{i=1}^{l}\frac{\|\mathbf{W}^i(s+1)(\mathbf{W}(s+1)^i)^\top-\mathbf{I}^l\|_F^2}{\|2\mathbf{W}^i(s)(\mathbf{W}^i(s))^\top-\mathbf{I}^l\|_F}\\
&+\lambda_d\sum_{j=1}^{d}\frac{\|\mathbf{W}_j(s+1)(\mathbf{W}_j(s+1))^\top-\mathbf{I}_d\|_F^2}{\|2\mathbf{W}_j(s)(\mathbf{W}_j(s))^\top-\mathbf{I}_d\|_F}\\
&+\lambda^{(c)}\sum_{k=1}^{c}\frac{\|\mathbf{W}^{(k)}(s+1)(\mathbf{W}^{(k)}(s+1))^\top-\mathbf{I}^{(c)}\|_F^2}{\|2\mathbf{W}^{(k)}(s)(\mathbf{W}^{(k)}(s))^\top-\mathbf{I}^{(c)}\|_F}\\
&\le\boldsymbol{\mathcal{F}}_1(s)+\boldsymbol{\mathcal{F}}_2(s)+\boldsymbol{\mathcal{F}}_3(s)+\lambda_2\sum_{k=1}^{c}\frac{\|\mathbf{U}^{(k)}(s)\|_F^2}{2\|\mathbf{U}^{(k)}(s)\|_F} \qquad (8.19)\\
&+\lambda_1\sum_{i=j}^{m}\sum_{k=1}^{c}\frac{\|\mathbf{V}_j^{(k)}(s)\mathbf{W}_j^{(k)}(s)\|_F^2}{2\|\mathbf{V}_j^{(k)}(s)\mathbf{W}_j^{(k)}(s)\|_F}\\
&+\lambda^l\sum_{i=1}^{l}\frac{\|\mathbf{W}^i(s)(\mathbf{W}(s)^i)^\top-\mathbf{I}^l\|_F^2}{\|2\mathbf{W}^i(s)(\mathbf{W}^i(s))^\top-\mathbf{I}^l\|_F}\\
&+\lambda_d\sum_{j=1}^{d}\frac{\|\mathbf{W}_j(s)(\mathbf{W}_j(s))^\top-\mathbf{I}_d\|_F^2}{\|2\mathbf{W}_j(s)(\mathbf{W}_j(s))^\top-\mathbf{I}_d\|_F}\\
&+\lambda^{(c)}\sum_{k=1}^{c}\frac{\|\mathbf{W}^{(k)}(s)(\mathbf{W}^{(k)}(s))^\top-\mathbf{I}^{(c)}\|_F^2}{\|2\mathbf{W}^{(k)}(s)(\mathbf{W}^{(k)}(s))^\top-\mathbf{I}^{(c)}\|_F}
\end{aligned}$$

Using Lemma 6, we obtain the following inequalities in Eqs. (8.20-8.24). From $\mathbf{Q}^B$ and $\mathbf{O}^B$, we obtain:

$$\begin{aligned}
&\sum_{i=1}^{m}\sum_{k=1}^{c}\Big(\|\mathbf{V}_j^{(k)}(s)\mathbf{W}_j^{(k)}(s+1)\|_F-\frac{\|\mathbf{V}_j^{(k)}(s)\mathbf{W}_j^{(k)}(s+1)\|_F^2}{2\|\mathbf{V}_j^{(k)}(s)\mathbf{W}_j^{(k)}(s)\|_F}\Big)\\
&\le\sum_{i=1}^{m}\sum_{k=1}^{c}\Big(\|\mathbf{V}_j^{(k)}(s)\mathbf{W}_j^{(k)}(s)\|_F-\frac{\|\mathbf{V}_j^{(k)}(s)\mathbf{W}_j^{(k)}(s)\|_F^2}{2\|\mathbf{V}_j^{(k)}(s)\mathbf{W}_j^{(k)}(s)\|_F}\Big) \qquad (8.20)
\end{aligned}$$

From $\mathbf{Q}^l$, we obtain:

$$\begin{aligned}&\sum_{i=1}^{m}\Big(\|\mathbf{W}^i(s+1)\mathbf{W}^{i\top}(s+1)-\mathbf{I}^l\|_F-\frac{\|\mathbf{W}^i(s+1)\mathbf{W}^{i\top}(s+1)-\mathbf{I}^l\|_F^2}{2\|\mathbf{W}^i(s)\mathbf{W}^{i\top}(s)-\mathbf{I}^l\|_F}\Big)\\&\leq\sum_{i=1}^{m}\Big(\|\mathbf{W}^i(s)\mathbf{W}^{i\top}(s)-\mathbf{I}^l\|_F-\frac{\|\mathbf{W}^i(s)\mathbf{W}^{i\top}(s)-\mathbf{I}^l\|_F^2}{2\|\mathbf{W}^i(s)(\mathbf{W}^{i\top}(s))-\mathbf{I}^l\|_F}\Big)\end{aligned}\tag{8.21}$$

From $\mathbf{Q}_d$, we obtain:

$$\begin{aligned}&\sum_{j=1}^{d}\Big(\|\mathbf{W}_j(s+1)(\mathbf{W}_j(s+1))^\top-\mathbf{I}_d\|_F-\frac{\|\mathbf{W}_j(s+1)(\mathbf{W}_j(s+1))^\top-\mathbf{I}_d\|_F^2}{2\|\mathbf{W}_j(s)(\mathbf{W}_j(s))^\top-\mathbf{I}_d\|_F}\Big)\\&\leq\sum_{j=1}^{d}\Big(\|\mathbf{W}_j(s)(\mathbf{W}_j(s))^\top-\mathbf{I}_d\|_F-\frac{\|\mathbf{W}_j(s)(\mathbf{W}_j(s))^\top-\mathbf{I}_d\|_F^2}{2\|\mathbf{W}_j(s)(\mathbf{W}_j(s))^\top-\mathbf{I}_d\|_F}\Big)\end{aligned}\tag{8.22}$$

From $\mathbf{Q}^{(c)}$, we obtain:

$$\begin{aligned}&\sum_{k=1}^{c}\Big(\|\mathbf{W}^{(k)}(s+1)(\mathbf{W}^{(k)}(s+1))^\top-\mathbf{I}^{(c)}\|_F-\\&\qquad\frac{\|\mathbf{W}^{(k)}(s+1)(\mathbf{W}^{(k)}(s+1))^\top-\mathbf{I}^{(c)}\|_F^2}{2\|\mathbf{W}^{(k)}(s)(\mathbf{W}^{(k)}(s))^\top-\mathbf{I}^{(c)}\|_F}\Big)\leq\\&\sum_{k=1}^{c}\Big(\|\mathbf{W}^{(k)}(s)(\mathbf{W}^{(k)}(s))^\top-\mathbf{I}^{(c)}\|_F-\frac{\|\mathbf{W}^{(k)}(s)(\mathbf{W}^{(k)}(s))^\top-\mathbf{I}^{(c)}\|_F^2}{2\|\mathbf{W}^{(k)}(s)(\mathbf{W}^{(k)}(s))^\top-\mathbf{I}^{(c)}\|_F}\Big)\end{aligned}\tag{8.23}$$

From $\mathbf{P}^{(R)}$, we obtain:

$$\begin{aligned}&\sum_{k=1}^{c}\Big(\|\mathbf{U}^{(k)}(s+1)\|_F-\frac{\|\mathbf{U}^{(k)}(s+1)\|_F^2}{2\mathbf{U}^{(k)}(s)\|_F}\Big)\\&\leq\sum_{k=1}^{c}\Big(\|\mathbf{U}^{(k)}(s)\|_F-\frac{\|\mathbf{U}^{(k)}(s)\|_F^2}{2\mathbf{U}^{(k)}(s)\|_F}\Big)\end{aligned}\tag{8.24}$$

Then, adding Eqs. (8.20-8.24) to Eq. (8.19) on both sides results in the following inequality:

$$
\begin{aligned}
&\boldsymbol{\mathcal{F}}_1(s+1) + \boldsymbol{\mathcal{F}}_2(s+1) + \boldsymbol{\mathcal{F}}_3(s+1) + \lambda_2 \sum_{k=1}^{c} \|\mathbf{U}^{(k)}(s+1)\|_F \\
&+ \lambda_1 \sum_{i=j}^{m} \sum_{k=1}^{c} \|\mathbf{V}_j^{(k)}(s)\mathbf{W}_j^{(k)}(s+1)\|_F \\
&+ \lambda^l \sum_{i=1}^{l} \|\mathbf{W}^i(s+1)\mathbf{W}^{i\top}(s+1) - \mathbf{I}^l\|_F \\
&+ \lambda_d \sum_{j=1}^{d} \|\mathbf{W}_j(s+1)\mathbf{W}_j^\top(s+1) - \mathbf{I}_d\|_F \\
&+ \lambda^{(c)} \sum_{k=1}^{c} \|\mathbf{W}^{(k)}(s+1)\mathbf{W}^{(k)\top}(s+1) - \mathbf{I}^{(c)}\|_F \\
&\leq \boldsymbol{\mathcal{F}}_1(s) + \boldsymbol{\mathcal{F}}_2(s) + \boldsymbol{\mathcal{F}}_3(s) + \lambda_2 \sum_{k=1}^{c} \|\mathbf{U}^{(k)}(s)\|_F \qquad (8.25) \\
&+ \lambda_1 \sum_{i=j}^{m} \sum_{k=1}^{c} \|\mathbf{V}_j^{(k)}(s)\mathbf{W}_j^{(k)}(s)\|_F \\
&+ \lambda^l \sum_{i=1}^{l} \|\mathbf{W}^i(s)\mathbf{W}^{i\top}(s) - \mathbf{I}^l\|_F + \lambda_d \sum_{j=1}^{d} \|\mathbf{W}_j(s)\mathbf{W}_j^\top(s) - \mathbf{I}_d\|_F \\
&+ \lambda^{(c)} \sum_{k=1}^{c} \|\mathbf{W}^{(k)}(s)\mathbf{W}^{(k)\top}(s) - \mathbf{I}^{(c)}\|_F
\end{aligned}
$$

This inequality in Eq. (8.25) proves that the objective value is decreased in each iteration. The Log-Cosh losses in Eq. (8.9) are convex [314] and the Frobenius norms in Eq. (8.9) are also convex [315]. Thus, as a sum of convex functions, the objective function in Eq. (8.9) is also convex. Therefore, Algorithm 6 is guaranteed to converge to the global optimal solution to the formulated constrained regularized optimization problem in Eq. (8.7). □

Time Complexity. As the formulated optimization problem in Eq. (8.7) is convex, Algorithm 7 converges fast (e.g., within tens of iterations only). In each iteration of our algorithm, computing Steps 3, 4, 5, 6, 8 and 10 is trivial. Steps 7, 9 and 11 can be computed by solving a system of linear equations with quadratic complexity.

8.5 Experiments

In this section, we first discuss the experimental setup and the implementation details of our approach. Then we analyze the experimental results obtained by our method and compare them with the previous state-of-the-art methods in multiple real-world off-road ground navigation scenarios.

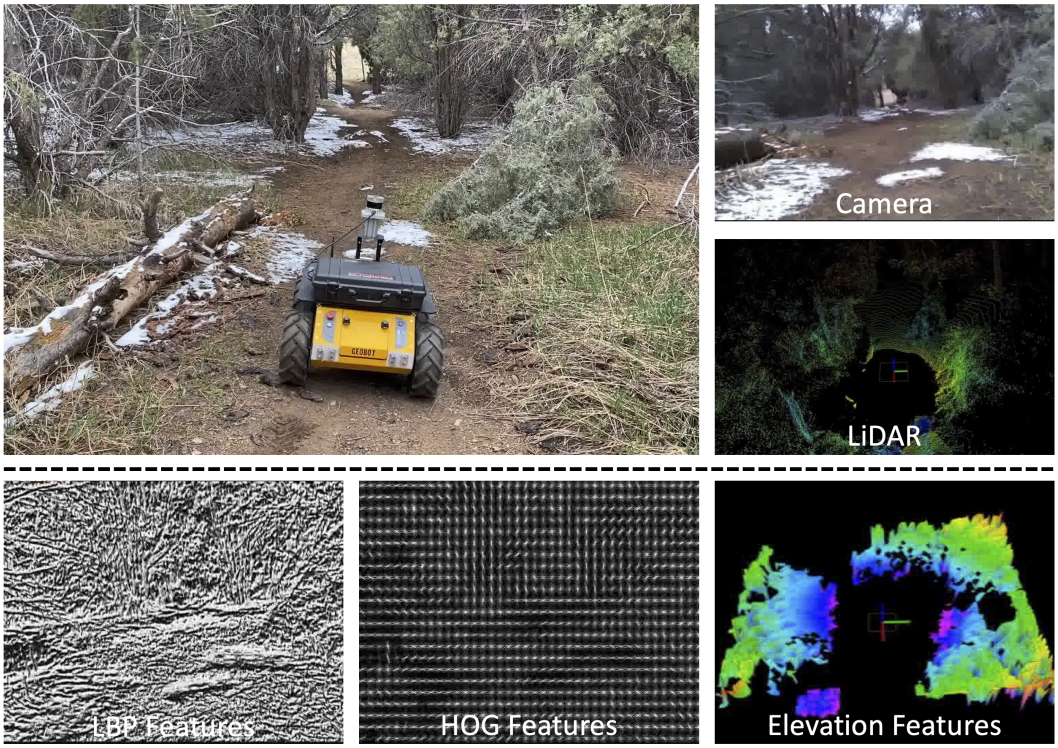


Figure 8.3 Multi-modal features extracted from the robot's onboard camera and LiDAR to characterize unstructured off-road terrains in our experiments.

8.5.1 Experimental Setups

We evaluate the proposed approach using Clearpath Husky ground robots that navigate on various unstructured off-road terrains. The robots are equipped with multiple exteroceptive and proprioceptive sensors. Exteroceptive sensors include an Intel Realsense D435 color-depth camera and an Ouster OS1-64 LiDAR to observe the terrain. Proprioceptive sensors include a Microstrain 3DM-GX5-25 Internal Measurement Unit (IMU) and wheel encoders, which measure the robot's states while navigating over these terrains. The observations are linearly interpolated to 15 Hz.

We implement multiple visual features that are extracted from color images to represent unstructured terrains, which include Histogram of Oriented Gradients (HOG) to represent the shape of the terrains [101] and Local Binary Patterns (LBP) to represent the texture of the terrain [152]. We also compute an elevation map [153] of the terrain round the robot using LiDAR data in order to represent the terrain characteristics, such as the slope, step height, and terrain normals. Moreover, we concatenate the IMU readings into a feature vector. The various features used in our experiments are demonstrated in Figure 8.3.

During training, the expected robot behaviors are recorded from expert demonstrations. The actual navigational behaviors are estimated using a SLAM technique [287]. We use a sequence of 5 past frames (i.e., $c = 5$) and set the hyperparameter values as $\lambda_1 = 1$ and $\lambda_2 = 0.1$ for all the experiments. We quantitatively evaluate our approach using four evaluation metrics, with a lower value indicating a better performance:

- *Failure Rate* is defined as the number of times that the robot fails to complete the navigation task within a set of experimental trails. A failure happens when the robot is stuck on the terrain or by an obstacle, or when the robot flips over.

- *Traversal Time* is defined as the average time for the robot to complete the navigation task over the given terrain.

- *Inconsistency* is defined as the difference between the robot's expected navigational behavior and its actual behavior in terms of the robot's poses.

- *Jerkiness* is defined as the average summation of the acceleration derivatives along all the axes. It indicates how smooth a robot can traverse a terrain. Reducing jerkiness is important to improve state estimation and SLAM that may assume smooth robot motions.

We compare the proposed approach with three previous robot navigation methods, including Model Predictive Path Integral (MPPI) [161], Non-Linear Control Optimization tool (NLOPT)

[316], and Terrain Representation and Apprenticeship Learning (TRAL) [6]. MPPI and NLOPT are control-based methods, whereas TRAL is a learning-based method.

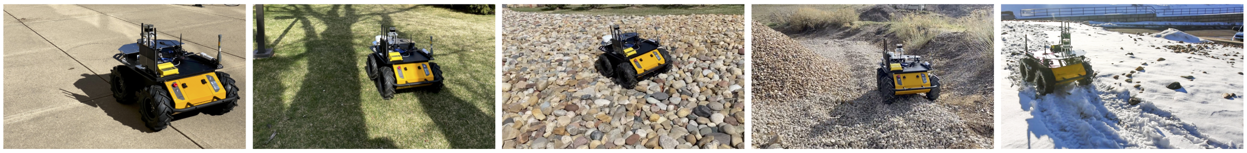


Figure 8.4 Individual types of terrains used in the experiments. From left to right: concrete, grass, large-rock, gravel, and snow.

Table 8.2 Quantitative results for scenarios when the robot traverses over the individual types of unstructured terrain shown in Figure 8.4. Successful runs (with no failures) are used to calculate the metrics of traversal time, inconsistency, and jerkiness. Our approach is compared against classical control-based approaches, MPPI [161] and NLOPT [316], and a learning-based method, TRAL [6].

	Failure Rate (/10)				Traversal Time (s)				Inconsistency				Jerkiness (m/s^3)			
Terrain	MPPI	NLOPT	TRAL	**Ours**	MPPI	NLOPT	TRAL	**Ours**	MPPI	NLOPT	TRAL	**Ours**	MPPI	NLOPT	TRAL	**Ours**
Concrete	0	0	0	0	12.82	17.82	16.04	12.49	0.65	0.79	0.69	0.56	5.13	4.98	5.19	4.35
Grass	0	0	0	0	13.45	18.02	16.39	13.32	0.61	0.90	0.88	0.62	7.82	8.52	8.32	6.98
Gravel	0	0	0	0	13.96	16.39	15.32	14.84	1.21	1.67	1.72	1.03	16.11	16.66	16.43	15.09
L.Rock	0	1	0	0	13.45	17.87	16.71	16.31	3.97	5.64	3.21	2.91	13.75	17.61	14.85	12.32
Snow	0	0	0	0	14.22	18.09	17.49	14.08	7.18	9.93	8.12	6.64	5.12	5.76	5.49	5.42

8.5.2 Robot Navigation over Individual Unstructured Off-road Terrains

In this set of experiments, the robot navigates over individual types of off-road terrains, including concrete, grass, gravel, large-rock, and snow. These terrains are illustrated in Figure 8.4. Each terrain track is around 10 meters long and has different characteristics. We train our method using the data collected when a human operator manually controls the robot to traverse over each of the terrains. The learned model is then deployed on the robot to generate terrain-aware controls as the ground robot autonomously navigates on the terrains. Evaluation metrics for each method are computed across ten trials on each type of individual terrain track.

The quantitative results achieved by our approach and its comparison with other methods are provided in Table 8.2. We observe that all methods enable the robot to successfully navigate over these individual types of off-road terrains with the exception of a single failure from NLOPT occurring on the large-rock terrain due to high centering. We compute the traversal time by averaging the time used by the robot to traverse over each terrain for all successful runs. We observe that our approach has the lowest traversal time over the concrete, grass, and snow terrains. For the gravel and large-rock terrains, our approach uses more time than MPPI. For all individual terrains, TRAL has a higher traversal time than MPPI and our approach, but NLOPT achieves the longest traversal time of all methods. Additionally, note that the model-based methods, MPPI and NLOPT, display the smallest standard deviations in traversal time across the set of terrains. This indicates that these approaches are not adapting their navigation approach to different terrain types. On the other hand, our approach is able to adaptively generate terrain-aware navigational controls. For example, our method generates a higher velocity control and obtains a lower traversal time over smoother terrains (e.g., concrete, grass, and snow).

Table 8.2 also shows the results of the inconsistency and jerkiness metrics. We observe that our approach achieves the lowest inconsistency for most individual terrains except the grass terrain, followed by the MPPI and TRAL methods. MPPI has the lowest inconsistency over grass terrain. TRAL obtains lower inconsistency than MPPI over the large-rock terrain, but performs worse than MPPI over other terrains. NLOPT performs the worst and has the highest inconsistency over all individual terrains. In general, we observe that the learning-based methods obtain a better performance than the model-based methods in terms of inconsistency. For the metric of jerkiness, we observe that our approach obtains the lowest jerkiness over most terrains except snow, where MPPI marginally outperforms our approach. For almost all cases, both NLOPT and TRAL obtain higher jerkiness values than MPPI and our approach across each terrain type.

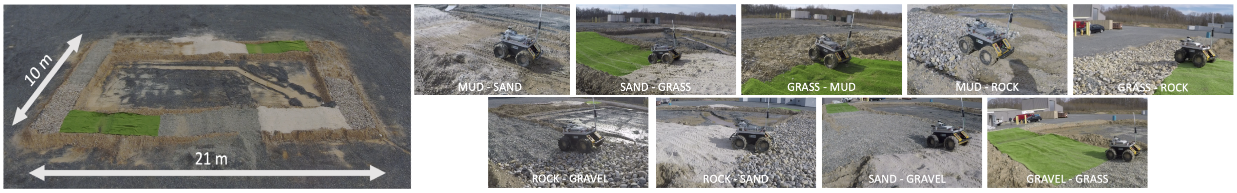


Figure 8.5 The off-road ground navigation circuit consists of different types of terrains (such as grass, gravel, large-rock, mud, and sand) as well as diverse terrain transitions. The size of the circuit is 10 x 21 meters with terrain slopes varying between $0 - 30°$.

Table 8.3 Quantitative results based on ten runs for scenarios when the robot navigates over the various terrain transitions shown in Figure 8.5. Successful runs (with no failures) are used to calculate the metrics of traversal time, inconsistency and jerkiness. Our approach is compared against classical control based approaches, MPPI [161] and NLOPT [316], and a learning-based method, TRAL [6].

	Failure Rate (/10)				Traversal Time (s)				Inconsistency				Jerkiness (m/s^3)			
Terrain	MPPI	NLOPT	TRAL	**Ours**	MPPI	NLOPT	TRAL	**Ours**	MPPI	NLOPT	TRAL	**Ours**	MPPI	NLOPT	TRAL	**Ours**
Mud-Sand	2	3	2	1	25.27	31.54	22.19	21.82	7.86	8.13	7.06	6.45	9.13	11.31	10.73	8.95
Sand-Grass	2	3	2	1	24.01	31.99	23.04	22.57	7.19	7.65	6.99	6.25	8.58	10.61	10.09	8.41
Grass-Mud	0	0	0	0	18.67	20.69	21.98	20.09	3.85	3.98	3.45	3.16	4.29	5.31	5.04	4.20
Mud-Rock	0	0	0	0	19.76	21.86	22.07	20.87	6.50	6.73	5.84	5.34	11.74	14.54	13.80	11.59
Rock-Sand	2	3	1	1	22.21	22.67	23.17	22.03	10.88	11.14	10.07	9.53	11.12	13.78	13.06	10.91
Sand-Gravel	2	3	1	1	23.09	23.98	23.65	22.98	11.81	12.28	10.61	9.70	15.57	19.28	18.30	15.26
Gravel-Grass	0	0	0	0	22.13	26.04	22.73	22.62	7.25	7.55	7.51	6.65	15.15	18.97	18.09	15.03
Grass-Rock	0	0	0	0	20.01	26.55	21.82	19.13	4.45	4.90	4.01	3.91	6.38	7.91	7.50	6.26
Rock-Gravel	0	0	0	0	17.89	27.03	22.41	20.17	3.78	3.91	4.31	3.18	13.34	16.53	15.68	13.09

8.5.3 Robot Adaptation to Terrain Transitions

In this set of experiments, we evaluate our proposed method when the ground robot navigates over unstructured terrain transitions. We leveraged an off-road ground navigation circuit, which is an experimental testing facility at the U.S. DEVCOM Army Research Laboratory Robotics Research Collaboration Campus (R2C2). An aerial view of the circuit and its terrain is shown in Figure 8.5.

We consider the scenario of a ground robot navigating the transitions between any combination of two terrains: grass, gravel, large rocks, mud, and sand. The aim is to evaluate how different approaches adapt with the changing terrains, e.g., robots need to slow down when transitioning from grass to rocks. In this experiment, no additional training is performed and the previously trained model from navigating over individual types of unstructured terrains is used. The evaluation metrics are computed from ten trial runs.

The quantitative results obtained by our approach and comparisons with other methods are provided in Table 8.3. In terms of failure rate, we observe that NLOPT has the worst performance. Comparatively, the MPPI and TRAL methods demonstrate a lower failure rate, with our approach achieving the lowest failure rate, thereby performing the best. Navigation failures for all methods are observed when the robot traverses over terrain combinations including sand. Failures while navigating over sand occur either when the robot gets high-centered or when the terrain slope is greater than 25 degrees. Our approach, with the ability to generate consistent behaviors, is able to compensate for wheel slip while traversing the sand terrain and obtains a lower failure rate.

In terms of the traversal time metric, we observe that both NLOPT and TRAL obtain the highest traversal time values, with NLOPT obtaining the highest value. Both MPPI and our approach obtain lower traversal time values. The traversal time values from our approach are higher when traversing terrains consisting of rock, sand, and mud, and are relatively lower when traversing terrains consisting of grass. This indicates the ability of our approach to adapt its behaviors to different types of terrains. The traversal time value is higher when navigating over terrains that include sand due to a higher rate of wheel slips.

In terms of the inconsistency metric, we observe that the NLOPT and TRAL methods obtain the highest values. Both MPPI and our approach have comparatively lower values of inconsistency, with our approach obtaining the smallest inconsistency values across terrain combinations. For all methods, the inconsistency value is high for terrain transitions including sand due to an increased rate of wheel slip. We observe a similar trend in terms of the jerkiness metric, where both the NLOPT and TRAL approaches obtain higher jerkiness values, with NLOPT having the highest value and performing the poorest. However, both MPPI and our approach obtain the least jerkiness value, with our approach performing slightly better than MPPI. For all methods, the inconsistency value is higher when traversing the terrain transitions including rock and gravel due to their rugged nature.

From this set of experiments, we observe that our approach generally performs the best over all the terrain combinations and outperforms previous methods as it achieves the fewest failures, while also performing best in terms of inconsistency and jerkiness.

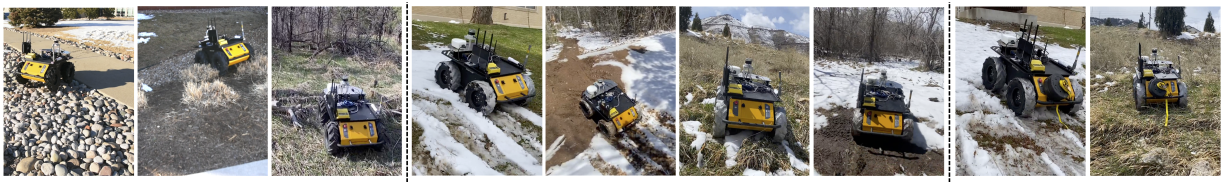


Figure 8.6 Various terrain tracks used to evaluate the self-reflection capability of the robot. From left to right: mixed terrain I (MT-I), mixed terrain II (MT-II), forest, reduced-traction snow-grass (RT-Snow-Grass), reduced-traction mud-snow (RT-Snow-Mud), reduced-traction hill (RT-Hill), reduced-traction forest (RT-Snow-Forest), reduced-traction payload snow-grass (RT-P-Snow-Grass) and reduced-traction payload hill terrain (RT-P-Hill).

8.5.4 Robot Self-Reflective Adaptation over Unstructured Terrains

In this set of experiments, we evaluate our approach's use of self-reflection to enhance robot navigation. Specifically, we perform experiments in scenarios where robots experience unexpected setbacks in their functionality. Each of these setbacks increases the difficulty of robot navigation over unstructured terrain and requires the robot to adapt via self-reflection. These setbacks are detailed below:

- *Over-Inflated Tires:* In this scenario, we over-inflate each tire on the wheeled ground robot, which causes reduced traction with the terrain. With this setback, the robot has a higher center of mass and increased jerkiness when traversing, as there is less damping from the over-inflated tires. We tested our approach with over-inflated tires on three different terrains. The first terrain, mixed terrain I (MT-I), is a mixture of concrete, large-rocks, grass and snow which the robot can experience them at once or individually. The second terrain, mixed terrain II (MT-II), is a mixture of grass, gravel, and bushes. Finally, the third terrain, forest terrain, is a cluttered environment with logs and rocks hidden by fallen leaves and tall grass. None of these three terrains have any significant slope.

- *Reduced Tire Traction:* In this scenario, we completely deflate the robot's tires and apply duct tape around the deflated tires, significantly reducing tire traction. This setback causes the robot to have a lower center of mass. We then evaluate our approach with reduced tire traction on four different terrains. The first terrain, snow-grass (referred to as

RT-Snow-Grass), is a grass terrain covered by snow. The second terrain, snow-mud (referred to as RT-Snow-Mud), is a mud terrain covered with snow. The third terrain, hill (referred to as RT-Hill), is a hill terrain consisting of rocks, snow, and logs covered by tall grass. Finally, the fourth terrain, snow-forest (referred to as RT-Snow-Forest), is a forest terrain covered with snow. Each of these terrains, with the exception of RT-Snow-Forest terrain, have varying slopes of $0°$– $45°$, which increases the difficulty of robot traversal.

- *Reduced Tire Traction with Increased Payload:* In this scenario, in addition to the deflated and duct-taped robot tires, we introduce 25 lbs of payload to the back of the robot. With this setback, we evaluate our approach on two different terrains. The first terrain, snow-grass (referred to as RT-P-Snow-Grass), is a grass-terrain covered with snow. The second terrain, hill (referred to as RT-P-Hill), consists of rocks and snow, covered by tall grass. Both these terrains have slopes ranging from $0°$– $45°$.

These nine different testing locations are demonstrated in Figure 8.6, which represent the challenges that a ground robot can experience in real-world unstructured environments. The evaluation metrics for each method are calculated based on ten trials of robot runs on each of the terrains. No additional training is performed, and we employ the same model from experiments on individual off-road terrains (from Section 8.5.2), with all parameters kept the same. The quantitative results obtained by our approach and its comparison with other approaches are presented in Table 8.3. We discuss the results of the three setback experiments next.

Over-Inflated Tires Results: For the scenario of over-inflated tires, we can observe that NLOPT obtains the highest failure rate. Both MPPI and TRAL perform better than NLOPT, achieving fewer failures. Comparatively, our approach achieves the lowest failure rate across all three terrains. In general, we observe that the failure rate is lower on the mixed terrain tracks, and higher on the cluttered forest terrain.

In terms of traversal time, we observe varying trends across the methods depending on the terrain type. Notice that NLOPT has the lowest traversal time on the forest terrain, but the highest traversal time on both the mixed terrains. On the other hand, our approach displays the inverse, with some of the lowest traversal times on the mixed terrain and the highest traversal time on the forest terrain. We see this increase in the traversal time on the forest terrain as an illustration of our approach's ability to adapt its navigational behaviors with cluttered terrains. That is, speed is reduced to ensure the robot can successfully navigate the environment. This is further supported by the fact that our approach achieves the fewest failure cases in the forest terrain. Also note that although NLOPT has the fastest traversal time for the forest terrain, it fails to reach the goal in half of the experimental trials. In general, MPPI and TRAL exhibit slower traversal rates than our approach, and in cases with faster traversal time we also observe that they result in higher failure rates.

In terms of the inconsistency metric, our approach, with its ability to generate consistent behaviors, obtains the lowest values and outperforms previous methods. Specifically, both TRAL and NLOPT have high inconsistency values over all terrains. In terms of the jerkiness metric, we observe that all approaches have high values in the forest terrain due to its cluttered nature. Our approach has the lowest jerkiness value in the mixed terrains but performs poorly in forest terrain. Across all three terrains, MPPI, NLOPT, and TRAL perform similarly with a slightly better performance exhibited from MPPI on two out of three terrain types.

Table 8.4 Quantitative results based on ten runs for the terrain scenarios shown in Figure 8.6 used to evaluate the self-reflection capability of the robot. Successful runs (with no failures) are used to calculate the metrics of traversal time, inconsistency and jerkiness. Our approach is compared against classical control based approaches, MPPI [161] and NLOPT [316], and a learning-based method, TRAL [6].

	Failure Rate (/10)				Traversal Time (s)				Inconsistency				Jerkiness (m/s^3)			
Terrain	MPPI	NLOPT	TRAL	**Ours**	MPPI	NLOPT	TRAL	**Ours**	MPPI	NLOPT	TRAL	**Ours**	MPPI	NLOPT	TRAL	**Ours**
MT-I	0	0	0	0	14.51	17.68	16.84	13.74	0.65	0.79	0.69	0.56	5.68	5.57	5.13	4.95
MT-II	1	2	1	0	13.44	17.48	16.74	13.64	0.78	0.98	0.88	0.71	7.66	8.75	8.35	6.52
Forest	2	5	3	1	14.67	13.67	15.29	15.96	1.21	1.67	1.72	1.11	16.25	16.62	16.55	16.78
RT-Snow-Grass	3	4	4	2	13.24	17.66	16.18	13.72	4.61	5.90	5.88	4.11	9.12	9.52	9.32	8.98
RT-Snow-Mud	5	7	4	2	13.28	17.84	16.10	12.03	3.97	5.64	3.21	3.01	13.14	17.15	14.12	12.35
RT-Hill	3	6	3	2	13.07	15.85	16.42	13.91	14.98	15.23	14.91	13.07	13.75	17.61	14.85	12.32
RT-Snow-Forest	0	0	0	0	13.55	16.63	14.39	14.34	7.18	9.93	8.12	6.62	6.72	6.88	6.92	6.47
RT-P-Snow-Grass	4	7	5	2	13.96	16.39	15.32	13.84	9.12	9.96	9.57	8.54	16.11	16.66	16.43	16.09
RT-P-Hill	5	8	6	4	14.59	15.87	14.38	15.77	11.18	15.93	14.12	10.62	13.12	17.56	14.86	12.51

Reduced Tire Traction Results: For the scenario of reduced-tire traction, generally we observe an increased failure rate compared to the scenario of over-inflated tires. Specifically, in the terrains with significant slopes, i.e., RT-Snow-Mud, RT-Hill, and RT-Snow-Grass, the failure rate is higher when compared to RT-Snow-Forest. Throughout all the terrains, we observe that the NLOPT approach obtains the highest failure rate and performs the poorest. Both MPPI and TRAL perform better, with our approach performing the best and obtaining the lowest failure rate.

In terms of the traversal time metric, we observe that the MPPI approach performs better than most approaches across all the terrains except for the RT-Snow-Mud terrain, where our approach has the lowest traversal time. This indicates the ability of our approach to exhibit terrain-aware behaviors as our approach is able to adapt its traversal speed with different terrains. Both TRAL and NLOPT have higher traversal times and exhibit poor performance over all terrains.

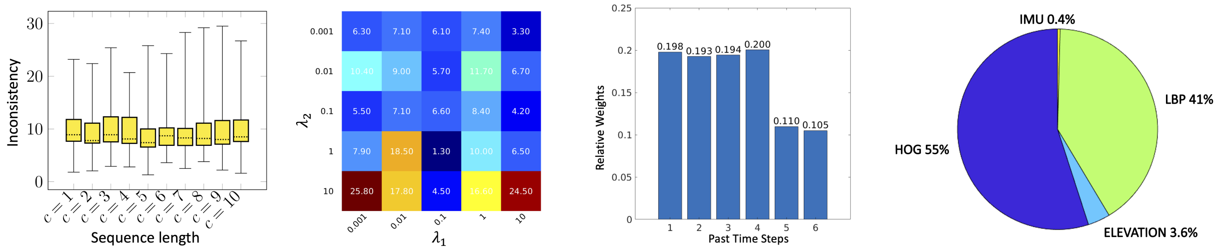
For the inconsistency metric, our approach obtains the lowest value over all terrains, followed by both MPPI and TRAL, with MPPI performing better on the RT-Snow-Grass and RT-Snow-Forest terrain, and TRAL performing better on the remaining terrains. The NLOPT approach obtains the highest inconsistency value, indicating a poor performance when compared to other approaches. When observing the jerkiness metrics obtained by the different approaches, we see a similar trend of our approach performing the best across different terrains, followed by both MPPI and TRAL. The NLOPT approach has some of the highest jerkiness values on all the terrains, indicating a poor performance overall. Overall, we see a smaller value of jerkiness from all methods on the RT-Snow-Forest terrain compared to the forest terrain discussed in the previous experiments with over-inflated tires. This indicates that the rugged nature of the forest terrain is suppressed when the terrain is covered with snow.

Reduced Tire Traction with Increased Payload Results: Finally, we evaluate our approach in the scenario of reduced tire traction with increased payload. These setbacks, especially on terrains with varying slopes, make it very challenging for mobile robots to traverse the terrain [317]. Accordingly, we observe that the failure rates are high in this scenario compared to the previous scenarios. Specifically, in the RT-P-Hill terrain, all methods fail at least 40% of the time. Overall, our approach achieves the best performance, followed by the MPPI, TRAL, and NLOPT approaches, respectively.

In terms of traversal time, we observe mixed results as our approach has the lowest traversal time on the RT-P-Snow-Grass terrain. However, it has the highest traversal time on the RT-P-Hill terrain, whereas the TRAL approach has the lowest traversal time. The MPPI approach has the second-best performance throughout, and the NLOPT approach has the highest traversal time on both terrains indicating a poor performance.

Similar to the previous scenarios, we observe that our approach has the lowest inconsistency value on both the terrains, followed by the MPPI and TRAL approaches. Again, we observe that the inconsistency values obtained by the NLOPT approach are higher than the other approaches. A similar trend is seen in terms of the jerkiness metric as well. Our approach has the best performance, followed by the MPPI approach on all terrains. Both the TRAL and NLOPT approaches have a higher value of jerkiness, indicating a poor performance.

Across all three categories of setbacks, it is observed that with the generation of consistent navigational behaviors, our approach can reduce the number of failures, with the trade-off of increased traversal time. Again, this increased traversal time can be attributed to the decrease in speed to ensure safety, which ultimately also helps produce lower inconsistency and jerkiness metrics.



(a) Frame Sequences

(b) Hyper-parameter Analysis

(c) Relative Importance of Time Steps

(d) Discriminative Feature Modalities

Figure 8.7 Analysis of our approach based on its various parameters.

8.5.5 Discussion

We further discuss the experimental results to investigate the characteristics of our approach. These investigations derive the basis for the various parameters used in our experimental runs. We include performance on various frame sequences, hyperparameter analysis, relative weights of each time step, and the discriminative feature modalities.

Dependence on Frame Sequences: Our approach uses a sequence of historical frames with length c to generate self-reflective consistent behaviors. Figure 8.7(a) illustrates the inconsistency value under different values of c. We use the best inconsistency value obtained from different runs of $\lambda_1 \in [0.001, 10]$ and $\lambda_2 \in [0.001, 10]$ for each c in this comparison. Figure 8.7(a) uses a box plot to display the distribution of results based on a five number summary (i.e., minimum, first quartlie, medium, thirds quartile, and maximum). Although, the performance of our approach barely differs over various frame sequence length c, the best performance is observed when $c = 5$, i.e., box plot with lowest minimum. This value of frame sequences is then used for all of our experiments. These values of c are affected by the robot's speed and can differ between robotic platforms.

Hyper-parameter Analysis: Our approach utilizes two hyper-parameters λ_1 and λ_2 that balance the amount of loss from our objective function and the regularization terms. Figure 8.7(b) depicts how the inconsistency metric changes with each of these hyperparameter values for five historical frame sequences, i.e., $c = 5$. In general, it is seen that the inconsistency metric changes significantly with variations in λ_2 as compared to λ_1, signifying the importance of modeling historical time steps. Altogether, it is observed that $\lambda_1 \in (0.01, 1)$ and $\lambda_2 \in (0.01, 10)$ result in a good performance. The best results are observed with $\lambda_1 = 0.1$ and $\lambda_2 = 1$. These are the fixed hyperparameters used during execution for all the experiments.

Relative Importance of Time Steps: In general, the best performance of our approach is observed when $c = 5$, meaning we leverage features from the current time step and the previous 5 time steps. Figure 8.7(c) further presents the importance of each of these time steps by displaying the relative weights of each. It is observed that the most recent time steps are more important when compared to earlier time steps. Specifically, the current and past three steps are equally important, with time step weightings of 20%. Whereas, the oldest time steps (i.e., fourth and fifth of $c = 5$) contribute a smaller importance of 10%. This breakdown further shows that features from previous time steps can provide equally important information and should be considered for generating self-reflective navigational behaviors.

Discriminative Feature Modalities: Our approach can automatically estimate the importance of the various feature modalities towards generating navigational behaviors. The results from evaluating the modality importance are shown in Figure 8.7(d). It is observed that two feature modalities, HOG and LBP, are most discriminative in performing self-reflective terrain adaptation. These two features, both extracted from camera images, combined account for 96% of the relative importance. The IMU features and elevation features from LiDAR are comparatively much less important. However, to perform self-reflection, we rely on the LiDAR data to perform SLAM [287] and provide the actual behaviors so the behavior differences can be computed. In this sense, LiDAR is also an important data modality for our approach.

8.6 Conclusion

In this paper, we introduce a novel method for self-reflective terrain-aware adaptation that enables an autonomous ground robot to generate consistent behaviors to navigate on unstructured off-road terrains. Formulated in a unified constrained regularized optimization framework, our approach monitors the inconsistency between expected and actual behaviors that is caused by robot setbacks, and then adapts the robot's navigational behaviors accordingly to enable consistent navigation. Also, our approach identifies discriminative terrain features and fuses them to perform effective adaptive navigational behaviors to changing terrain. We design an algorithm to solve the formulated optimization problem, which holds a theoretical guarantee to converge to the global optimal solution. To evaluate our approach, we conduct extensive experiments using real ground robots with varying functionality changes over diverse unstructured off-road terrains. Experimental results have shown that our self-reflective terrain-aware adaptation method outperforms previous and baseline techniques and enables ground robots to generate consistent behaviors when navigating in off-road environments.

CHAPTER 9
CONCLUSION

This dissertation addresses robot learning for lifelong adaptation in open-world environments, focusing on enabling robots to be intelligently adjust and continously improve under long-term challenges enabling robust, adaptive and resilient autonomy. I summarize the key contributions below to conclude this dissertation.

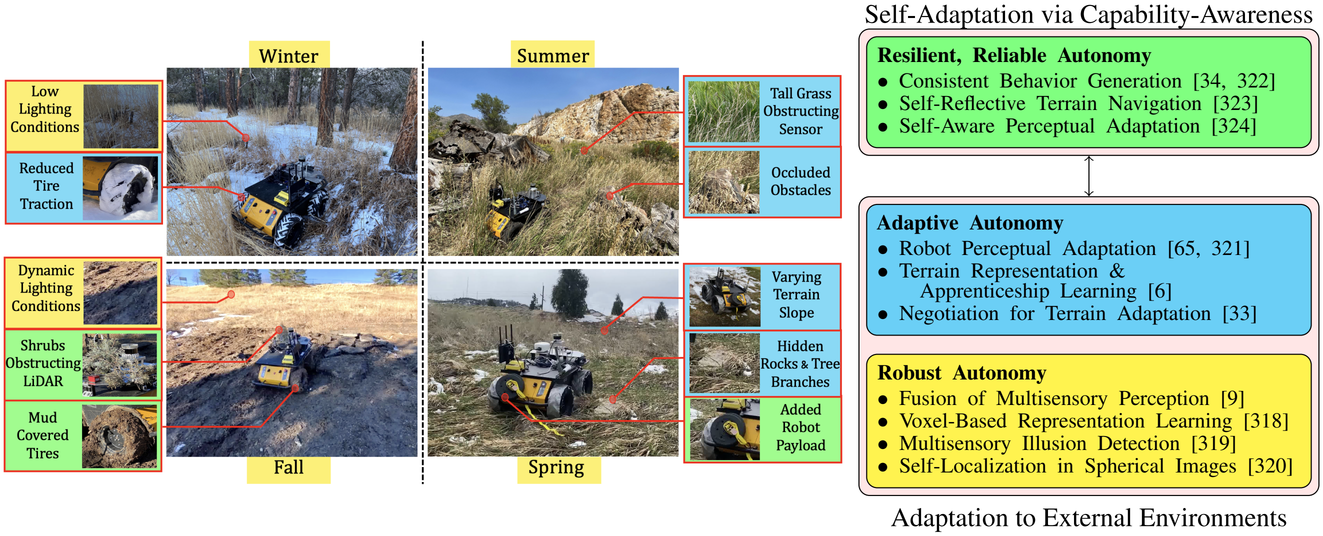

Winter
Summer
Fall
Spring
Low Lighting Conditions
Reduced Tire Traction
Dynamic Lighting Conditions
Shrubs Obstructing LiDAR
Mud Covered Tires
Tall Grass Obstructing Sensor
Occluded Obstacles
Varying Terrain Slope
Hidden Rocks & Tree Branches
Added Robot Payload
Self-Adaptation via Capability-Awareness
Resilient, Reliable Autonomy
• Consistent Behavior Generation [34, 322]
• Self-Reflective Terrain Navigation [323]
• Self-Aware Perceptual Adaptation [324]
Adaptive Autonomy
• Robot Perceptual Adaptation [65, 321]
• Terrain Representation & Apprenticeship Learning [6]
• Negotiation for Terrain Adaptation [33]
Robust Autonomy
• Fusion of Multisensory Perception [9]
• Voxel-Based Representation Learning [318]
• Multisensory Illusion Detection [319]
• Self-Localization in Spherical Images [320]
Adaptation to External Environments

Figure 9.1 Research accomplishments on robot learning for lifelong adaptation in open-world environments.

1. I propose a new problem of place recognition from omnidirectional multisensory observations in long-term autonomy. To address this challenge, I introduce the novel FOMP approach that is able to identify and integrate discriminative multimodal data obtained from heterogeneous sensors in different views. My approach shows that different viewing angles in the omnidirectional observation have different description powers. My research also demonstrates that multi-directional long-term place recognition is achievable. To validate my FOMP approach and hypothesis, I collected a large-scale dataset containing omnidirectional multisensory observations. Experimental results on this dataset have demonstrated that FOMP obtains promising long-term place recognition performance.

2. I introduced the novel, bio-inspired approach named RObot Perceptual Adaptation (ROPA) to enable the new robot capability of calibrating multisensory perception, in order for robots to adapt to short-term and long-term environment changes. The focused application in this paper aims at the task of long-term human following in the field environment, which is essential to real-world human-robot teaming applications. The ROPA approach is formulated as a joint learning problem to simultaneously estimate the representativeness of each feature modalities, integrate heterogeneous features, and more importantly, calibrate the perception model to adapt multisensory perception with environmental changes. In order to fuse multisensory input data, I implement sparsity-inducing norms that enforces the base perception model to learn sparse weights of the multisensory input features and use the learned weights for multisensory fusion. To achieve perception calibration, I estimate the representativeness of the input features to encode the environment, and provide a calibration of the base model according to the environmental change. All components are integrated under the unified theoretical framework of regularized optimization. I conducted extensive experiments to evaluate and analyze ROPA in various scenarios using PEAC dataset. Experiment results have validated that, through calibrating perception, ROPA is able to effectively adapt to environment changes, obtains promising accuracy and efficiency, and outperforms baseline and previous methods in dynamic and unstructured environments in long-term human following applications.

3. I presented the Terrain Representation and Apprenticeship Learning (TRAL) approach to address the research problem of robot adaptation to unstructured terrains. As a theoretical novelty, TRAL formulates representation and apprenticeship learning in the unified regularized optimization framework to perform joint robot learning for terrain classification and behavior generation, which also automatically estimates the importance of terrain feature modalities. Our second theoretical novelty is the design of an optimization algorithm to solve the formulated problem, which possesses a theoretical convergence guarantee. The proposed approach is extensively evaluated using two real-world scenarios in the experiments, in which

a mobile robot navigates over familiar or unfamiliar unstructured terrains. Experimental results have demonstrated that TRAL is able to transfer human expertise to mobile robots with small errors, obtain superior performance compared with previous and baseline methods, and offer intuitive insights into the importance of terrain feature modalities.

4. I introduce the novel Negotiation for Adaptation over Unstructured Terrain Surfaces (NAUTS) approach for robot adaptation by negotiation for navigating in unstructured terrains, which enables ground robots to adapt their navigation policies using a negotiation process. Our approach learns a prediction model to function as a terrain-aware joint local controller and planner corresponding to various policies and then uses the negotiation process to form agreements between these policies in order to improve robot navigation efficiency. Moreover, our approach explores different policies to improve navigation efficiency in a given environment continuously. We also developed a novel optimization algorithm that solves an optimal solution to the robot negotiation problem in real time. Experimental results have shown that our approach enables a robot to negotiate its behaviors with the terrain and delivers more successful and efficient navigation compared to previous methods.

5. I introduce a novel approach for consistent behavior generation that enables ground robots' actual behavior to more accurately match expected behaviors while adapting to a variety of unstructured off-road terrain. My approach learns offset behaviors in a self-supervised fashion to compensate for the inconsistency between the actual and expected navigational behaviors without the need to explicitly model various setbacks, and learns the importance of the multi-modal features to improve the representation of terrain for better adaptation. This proposed approach is extensively evaluated in real-world off-road environments. Experimental results have shown that the proposed approach enables a robot to improve its ground maneuverability when traversing over complex unstructured off-road terrain with more behavior consistency and smoothness compared to previous methods.

6. I introduce a novel human-inspired approach to learning capability-aware perceptual adaptation, which allows the robot to adjust to perceptual challenges and successfully navigate in unstructured off-road environments. Our approach learns capability awareness in perception through the learning of a modality invariant trajectory representation that allows for encoding a common representation of information from all the sensors in a much smaller observation space in a self-supervised fashion. To evaluate our approach, we perform extensive experiments using a physical ground robot with real-world scenarios of different perceptual challenges over various unstructured off-road terrains. Experimental results prove that our method can better adapt to these perceptual challenges when navigating over unstructured terrains and outperform previous and baseline methods.

7. I introduce a novel method for self-reflective terrain-aware adaptation that enables an autonomous ground robot to generate consistent behaviors to navigate on unstructured off-road terrains. Formulated in a unified constrained regularized optimization framework, our approach monitors the inconsistency between expected and actual behaviors that are caused by robot setbacks and then adapts the robot's navigational behaviors accordingly to enable consistent navigation. Also, our approach identifies discriminative terrain features and fuses them to perform effective adaptive navigational behaviors to changing terrain. We design an algorithm to solve the formulated optimization problem, which holds a theoretical guarantee to converge to the global optimal solution. To evaluate our approach, we conduct extensive experiments using real ground robots with varying functionality changes over diverse unstructured off-road terrains. Experimental results have shown that our self-reflective terrain-aware adaptation method outperforms previous and baseline techniques and enables ground robots to generate consistent behaviors when navigating in off-road environments.

9.1 Future Directions

Building on the work completed in this dissertation, other research problems can be further studied as an extension of this dissertation.

9.1.1 Semantic Aware Active Perception for Enhanced Terrain Awareness

By using Semantic Reasoning techniques, robotic systems can encode and leverage semantic knowledge such as concepts, facts, ideas, and beliefs about the world to enhance their understanding of the environment. Continually perceiving, understanding, and generalizing semantic knowledge allows a robot to identify the meaningful patterns shared between problems and environments, and therefore more effectively adapt to novel situations, better navigate in complex environments, and perform a wide range of real-world tasks in general. The goal of this research is to enhance multisensory perception for semantic understanding of navigation-relevant objects such as terrain, vehicles, and adversaries. Specifically, the focus is on improving perception under conditions of uncertainty, while also developing methods for generating local navigational controls that optimize the robot's motion and minimize perceptual uncertainty simultaneously. To achieve this, a two-stage process can be used. First, use Bayesian deep networks to perform joint inference, allowing the robot to learn representations that account for both stochastic (aleatoric) and systematic (epistemic) uncertainties in its perception of the environment. Second, develop and extend soft actor-critic reinforcement learning, leveraging the learned representations and uncertainty estimates to generate optimal navigational controls and plans for the goal position. By accurately capturing the underlying statistical structure of the robot's perception, these representations will enable accurate understanding of the world and efficient navigational control that can be tailored to specific environmental conditions and objectives.

9.1.2 Proactive High-Speed Autonomy

Proactive adaptation enables robots to predict the outcome of their behaviors to impact themselves in an environment and adjust their behaviors to optimize their future performance. Building on previous research, I will investigate learning of robot-terrain interactions at high speeds and use this information to generate more efficient navigational controls and plans while reducing navigational failures and increasing efficiency. The key idea is to adaptively learn the robot-terrain interaction dynamics model and using it to continuously predict traversability costmaps in unstructured environments. In a self-supervised fashion, the robot learns to predict its future terrain interaction based exteroceptive sensory information, uncertainty estimates, and finally use feedback from proprioceptive sensors to update terrain interactions dynamics model. An alternative approach is adversarial costmap learning, where a generator implicitly models robot-terrain interactions during high-speed navigation to generate robot-centric costmaps, and the discriminator uses proprioceptive sensor feedback to identify incorrect cost representations. This approach leverages high frequency update of color cameras to improve high-speed planning when LiDAR sensors have limitations and would also greatly reduce computational overhead of generating costmaps.

9.1.3 Heterogenous Multi-Robot Collaborative Autonomy

The goal of this research is to develop self-reliant cooperative dynamic teams of heterogenous robots that operate with increased speed, increased reliability, improved redundancy, enhanced flexibility, and expanded operational reach. Through my research, I aim to provide a more comprehensive and higher-confidence perceptual understanding in dynamic and uncertain environments through multi-robot collaborative semantic reasoning. This capability is highly relevant to teams of robots performing reconnaissance and surveillance missions, as they can search for potential threats by analyzing the environment and behaviors of individuals, thus improving overall situational awareness. Two primary challenges will be addressed to achieve this objective. Firstly, identifying whether multiple robots are observing the same activity or scene in highly dynamic and contested environments, is challenging due to the different perspectives of robots and the uncertainty inherent in these environments. Secondly, reaching a consensus among heterogeneous robots based on their uncertainties of reasoning results, which is essential to integrate the information and reasoning results from multiple robots to achieve higher confidence and comprehensive perceptual understanding.

REFERENCES

[1] Yasuyoshi Yokokohji. The Use of Robots to Respond to Nuclear Accidents: Applying the Lessons of the Past to the Fukushima Daiichi Nuclear Power Station. *Annual Review of Control, Robotics, and Autonomous Systems*, 2021.

[2] Mauro Dimastrogiovanni, Florian Cordes, and Giulio Reina. Terrain Estimation for Planetary Exploration Robots. *Applied Sciences*, 2020.

[3] Rajiv Kumar Modanval, S Rakesh Kumar, N Gayathri, and Sandeep Kr Sharma. Robotics, AI and IoT in the Defense Sector. *AI and IoT-Based Intelligent Automation in Robotics*, 2021.

[4] Eric R Teoh and David G Kidd. Rage Against the Machine? Google's Self-Driving Cars Versus Human Drivers. *Journal of safety research*, 2017.

[5] Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Ving: Learning Open-World Navigation with Visual Goals. In *International Conference on Robotics and Automation (ICRA)*, 2021.

[6] Sriram Siva, Maggie Wigness, John Rogers, and Hao Zhang. Robot adaptation to unstructured terrains by joint representation and apprenticeship learning. In *Robotics: Science and Systems*, 2019.

[7] Yuqian Jiang, Nick Walker, Justin Hart, and Peter Stone. Open-World Reasoning for Service Robots. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, pages 725–733, 2019.

[8] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement Learning in Robotics: A Survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[9] Sriram Siva and Hao Zhang. Omnidirectional Multisensory Perception Fusion for Long-Term Place Recognition. In *International Conference on Robotics and Automation*, 2018.

[10] Sriram Siva and Hao Zhang. Robot Perceptual Adaptation to Environment Changes for Long-Term Human Teammate Following. *The International Journal of Robotics Research*, page 0278364919896625, 2022.

[11] Jim Pugh, Alcherio Martinoli, and Yizhen Zhang. Particle Swarm Optimization for Unsupervised Robotic Learning. In *Proceedings of IEEE Swarm Intelligence Symposium*, 2005.

[12] Chenxia Wu, Jiemi Zhang, Ozan Sener, Bart Selman, Silvio Savarese, and Ashutosh Saxena. Watch-N-Patch: Unsupervised Learning of Actions and Relations. *Transactions on Pattern Analysis and Machine Intelligence*, pages 467–481, 2017.

[13] Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. How to Train your Robot with Deep Reinforcement Learning: Lessons we have Learned. *The International Journal of Robotics Research*, 40(4-5):698–721, 2021.

[14] Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5: 411–444, 2022.

[15] Mouhacine Benosman. *Learning-based adaptive control: An extremum seeking approach–theory and applications*. Butterworth-Heinemann, 2016.

[16] Michael W Floyd, Michael Drinkwater, and David W Aha. Trust-guided behavior adaptation using case-based reasoning. Technical report, Naval Research Laboratory Washington United States, 2015.

[17] Qiqian Zhang, Hui Qian, and Miaoliang Zhu. Parameter adaptation by case-based mission-planning of outdoor autonomous mobile robot. In *Proceedings of Intelligent Vehicles Symposium*, 2005.

[18] Karl Johan Åström and Peter Eykhoff. System identification—a survey. *Automatica*, 7(2): 123–162, 1971.

[19] EM Wright. Adaptive control processes: a guided tour. by richard bellman. 1961. 42s. pp. xvi+ 255.(princeton university press). *The Mathematical Gazette*, 46(356):160–161, 1962.

[20] Yakov Zalmanovich Tsypkin and Zivorad Jezdimir Nikolic. *Adaptation and learning in automatic systems*, volume 73. Academic Press New York, 1971.

[21] Alexander Dettmann, Malte Langosz, Kai von Szadkowski, and Sebastian Bartsch. Towards lifelong learning of optimal control for kinematically complex robots. In *Workshop at IEEE International Conference on Robotics and Automation*, 2014.

[22] Noriaki Mitsunaga, Christian Smith, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. Robot behavior adaptation for human-robot interaction based on policy gradient reinforcement learning. *Journal of the Robotics Society of Japan*, 24(7):820–829, 2006.

[23] Hannes Ritschel and Elisabeth André. Real-time robot personality adaptation based on reinforcement learning and social signals. In *ACM/IEEE International Conference on Human-Robot Interaction*, 2017.

[24] Aleksandar Jevtić, Adria Colomé, Guillem Alenya, and Carme Torras. Robot motion adaptation through user intervention and reinforcement learning. *Pattern Recognition Letters*, 2017.

[25] Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to Adapt in Dynamic, Real-World Environments Through Meta-Reinforcement Learning. *ICLR*, 2019.

[26] Chao Lu, Fengqing Hu, Dongpu Cao, Jianwei Gong, Yang Xing, and Zirui Li. Transfer Learning for Driver Model Adaptation in Lane-Changing Scenarios using Manifold Alignment. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[27] Yueyue Liu, Zhijun Li, Huaping Liu, and Zhen Kan. Skill Transfer Learning for Autonomous Robots and Human-Robot Cooperation: A Survey. *Robotics and Autonomous Systems*, 2020.

[28] Yu Sun, Wyatt L Ubellacker, Wen-Loong Ma, Xiang Zhang, Changhao Wang, Noel V Csomay-Shanklin, Masayoshi Tomizuka, Koushil Sreenath, and Aaron D Ames. Online Learning of Unknown Dynamics for Model-Based Controllers in Legged Locomotion. *Robotics and Automation Letters*, 6(4):8442–8449, 2021.

[29] Zhi Qiang Tang, Ho Lam Heung, Kai Yu Tong, and Zheng Li. Model-Based Online Learning and Adaptive Control for a "Human-Wearable Soft Robot" Integrated System. *The International Journal of Robotics Research*, 40(1):256–276, 2021.

[30] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming Catastrophic Forgetting with Hard Attention to the Task. In *ICML*, 2018.

[31] Crystal Chao, Maya Cakmak, and Andrea L Thomaz. Transparent Active Learning for Robots. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2010.

[32] Chao Guo, Tianxiang Bai, Xiao Wang, Xiangyu Zhang, Yue Lu, Xingyuan Dai, and Fei-Yue Wang. ShadowPainter: Active Learning enabled Robotic Painting through Visual Measurement and Reproduction of the Artistic Creation Process. *Journal of Intelligent & Robotic Systems*, 2022.

[33] Sriram Siva, Maggie Wigness, John G Rogers, Long Quang, and Hao Zhang. NAUTS: Negotiation for Adaptation to Unstructured Terrain Surfaces. *International Conference on Intelligent Robots and Systems*, 2022.

[34] Sriram Siva, Maggie Wigness, John Rogers, and Hao Zhang. Enhancing Consistent Ground Maneuverability by Robot Adaptation to Complex Off-Road Terrains. In *CoRL*, 2021.

[35] Michael J Milford and Gordon F Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *International conference on Robotics and Automation*, 2012.

[36] Fei Han, Hua Wang, and Hao Zhang. Learning of integrated holism-landmark representations for long-term loop closure detection. In *Association for the Advancement of Artificial Intelligence*, 2018.

[37] Tayyab Naseer, Luciano Spinello, Wolfram Burgard, and Cyrill Stachniss. Robust visual robot localization across seasons using network flows. In *Conference on Artificial Intelligence*, 2014.

[38] Aude Oliva and Antonio Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research*, 155:23–36, 2006.

[39] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Conference on Computer Vision and Pattern Recognition*, 2005.

[40] Roberto Arroyo, Pablo F Alcantarilla, Luis M Bergasa, and Eduardo Romera. Towards life-long visual localization using an efficient matching of binary sequences from images. In *International Conference on Robotics and Automation*, 2015.

[41] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features-SURF. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[42] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-SLAM: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[43] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016.

[44] Mathieu Labbe and Francois Michaud. Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Transactions on Robotics*, 29(3):734–745, 2013.

[45] Mark Cummins and Paul Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[46] Michael J Milford, Gordon F Wyeth, and David Prasser. Rat-SLAM: a hippocampal model for simultaneous localization and mapping. In *International conference on Robotics and Automation*, 2004.

[47] Hao Zhang, Fei Han, and Hua Wang. Robust multimodal sequence-based loop closure detection via structured sparsity. In *Robotics: Science and Systems*, 2016.

[48] Cheng Chen and Han Wang. Appearance-based topological bayesian inference for loop-closing detection in a cross-country environment. *The International Journal of Robotics Research*, 25(10):953–983, 2006.

[49] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[50] Niko Sunderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems*, 2015.

[51] Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford. On the performance of convnet features for place recognition. In *Intelligent Robots and Systems*, 2015.

[52] Zetao Chen, Obadiah Lam, Adam Jacobson, and Michael Milford. Convolutional neural network-based place recognition. *Australian Conference of Robotics and Automation*, 2014.

[53] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Conference on Computer Vision and Pattern Recognition*, 2013.

[54] Fei Han, Xue Yang, Yiming Deng, Mark Rentschler, Dejun Yang, and Hao Zhang. SRAL: Shared representative appearance learning for long-term visual place recognition. *IEEE Robotics and Automation Letters*, 2(2):1172–1179, 2017.

[55] Irina F Gorodnitsky and Bhaskar D Rao. Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm. *IEEE Transactions on signal processing*, 45(3):600–616, 1997.

[56] Jianxin Wu and Jim M Rehg. Centrist: A visual descriptor for scene categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1489–1501, 2011.

[57] Lin Yuan, Kai Chi Chan, and CS George Lee. Robust semantic place recognition with vocabulary tree and landmark detection. In *IROS workshop on Active Semantic Perception and Object Search in the Real, World*, 2011.

[58] Ian Watson and Farhi Marir. Case-based reasoning: A review. *The knowledge Engineering Review*, 9(4):327–354, 1994.

[59] Hani Hagras, Victor Callaghan, and Martin Colley. Learning and adaptation of an intelligent mobile robot navigator operating in unstructured environment based on a novel online fuzzy–genetic system. *Fuzzy Sets and Systems*, 141(1):107–160, 2004.

[60] Adriana Tapus, Cristian Tapus, and Maja Matarić. Long term learning and online robot behavior adaptation for individuals with physical and cognitive impairments. In *International Conference on Field and Service Robotics*, 2010.

[61] Changchun Liu, Karla Conn, Nilanjan Sarkar, and Wendy Stone. Online affect detection and robot behavior adaptation for intervention of children with autism. *IEEE transactions on robotics*, 24(4):883–896, 2008.

[62] Gillian Rhodes, Tamara L Watson, Linda Jeffery, and Colin WG Clifford. Perceptual adaptation helps us identify faces. *Vision Research*, 50(10):963–968, 2010.

[63] Maria Vinas, Lucie Sawides, Pablo De Gracia, and Susana Marcos. Perceptual adaptation to the correction of natural astigmatism. *PLoS One*, 7(9):e46361, 2012.

[64] Michael A Webster. Adaptation and visual coding. *Journal of vision*, 11(5):3–3, 2011.

[65] Sriram Siva and Hao Zhang. Robot adaptation to environment changes in long-term autonomy. 2018.

[66] Alexander Kleiner, Markus Dietl, and Bernhard Nebel. Towards a life-long learning soccer agent. In *Robot Soccer World Cup*, 2002.

[67] Sebastian Thrun and Tom M Mitchell. Lifelong robot learning. In *The Biology and Technology of Intelligent Autonomous Agents*, pages 165–196. 1995.

[68] Adam White, Joseph Modayil, and Richard S Sutton. Scaling life-long off-policy learning. In *IEEE International Conference on Development and Learning and Epigenetic Robotics*, 2012.

[69] Iolanda Leite, Carlos Martinho, and Ana Paiva. Social robots for long-term interaction: a survey. *International Journal of Social Robotics*, 5(2):291–308, 2013.

[70] Takayuki Kanda, Masahiro Shiomi, Zenta Miyashita, Hiroshi Ishiguro, and Norihiro Hagita. A communication robot in a shopping mall. *IEEE Transactions on Robotics*, 26(5):897–913, 2010.

[71] Ylva Fernaeus, Maria Håkansson, Mattias Jacobsson, and Sara Ljungblad. How do you play with a robotic toy animal?: a long-term study of pleo. In *Proceedings of International Conference on Interaction Design and Children*, 2010.

[72] David M Rosen, Julian Mason, and John J Leonard. Towards lifelong feature-based mapping in semi-static environments. In *IEEE International Conference on Robotics and Automation*, 2016.

[73] Stephanie M Lowry, Michael J Milford, and Gordon F Wyeth. Transforming morning to afternoon using linear regression techniques. In *IEEE International Conference on Robotics and Automation*, 2014.

[74] Peer Neubert, Niko Sünderhauf, and Peter Protzel. Appearance change prediction for long-term navigation across seasons. In *European Conference on Mobile Robots*, 2013.

[75] Niko Sünderhauf, Peer Neubert, and Peter Protzel. Predicting the change–a step towards life-long operation in everyday environments. *IEEE Workshop on Robotics Challenges and Vision*, 2014.

[76] Roberto Arroyo, Pablo F Alcantarilla, Luis M Bergasa, and Eduardo Romera. Fusion and binarization of CNN features for robust topological localization across seasons. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.

[77] Fei Han, Xue Yang, Yiming Deng, Mark Rentschler, Dejun Yang, and Hao Zhang. Life-long place recognition by shared representative appearance learning. 2016.

[78] Caleb Tung, Matthew R Kelleher, Ryan J Schlueter, Binhan Xu, Yung-Hsiang Lu, George K Thiruvathukal, Yen-Kuang Chen, and Yang Lu. Large-scale object detection of images from network cameras in variable ambient lighting conditions. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 393–398. IEEE, 2019.

[79] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[80] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[81] Sebastian Thrun, Maren Bennewitz, Wolfram Burgard, Armin B Cremers, Frank Dellaert, Dieter Fox, Dirk Hahnel, Charles Rosenberg, Nicholas Roy, Jamieson Schulte, et al. Minerva: A second-generation museum tour-guide robot. In *IEEE International Conference on Robotics and Automation*, 1999.

[82] Takanori Shibata and Kazuo Tanie. Influence of a priori knowledge in subjective interpretation and evaluation by short-term interaction with mental commit robot. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000.

[83] Lynne E Parker. Lifelong adaptation in heterogeneous multi-robot teams: Response to continual variation in individual tobot performance. *Autonomous Robots*, 8(3):239–267, 2000.

[84] Kazuhiro Kosuge and Norihide Kazamura. Control of a robot handling an object in cooperation with a human. In *IEEE International Workshop on Robot and Human Communication*, 1997.

[85] Paul Evrard, Elena Gribovskaya, Sylvain Calinon, Aude Billard, and Abderrahmane Kheddar. Teaching physical collaborative tasks: object-lifting case study with a humanoid. In *IEEE-RAS International Conference on Humanoid Robots*, 2009.

[86] Elena Gribovskaya, Abderrahmane Kheddar, and Aude Billard. Motion learning and adaptive impedance for robot control during physical interaction with humans. In *IEEE International Conference on Robotics and Automation*, 2011.

[87] Don Joven Agravante, Andrea Cherubini, Antoine Bussy, Pierre Gergondet, and Abderrahmane Kheddar. Collaborative human-humanoid carrying using vision and haptic sensing. In *IEEE International Conference on Robotics and Automation*, 2014.

[88] José Ramón Medina, Michael Shelley, Dongheui Lee, Wataru Takano, and Sandra Hirche. Towards interactive physical robotic assistance: Parameterizing motion primitives through natural language. In *IEEE International Conference on Robot and Human Interactive Communication*, 2012.

[89] Stefanos Nikolaidis, Yu Xiang Zhu, David Hsu, and Siddhartha Srinivasa. Human-robot mutual adaptation in shared autonomy. In *ACM/IEEE International Conference on Human-Robot Interaction*, 2017.

[90] Stefanos Nikolaidis, Swaprava Nath, Ariel D Procaccia, and Siddhartha Srinivasa. Game-theoretic modeling of human adaptation in human-robot collaboration. In *Proceedings of ACM/IEEE International Conference on Human-Robot Interaction*, 2017.

[91] Yanan Li, Keng Peng Tee, Wei Liang Chan, Rui Yan, Yuanwei Chua, and Dilip Kumar Limbu. Continuous role adaptation for human–robot shared control. *IEEE Transactions on Robotics*, 31(3):672–681, 2015.

[92] Ivana Kruijff-Korbayová, Francis Colas, Mario Gianni, Fiora Pirri, Joachim de Greeff, Koen Hindriks, Mark Neerincx, Petter Ögren, Tomáš Svoboda, and Rainer Worst. Tradr project: Long-term human-robot teaming for robot assisted disaster response. *KI-Künstliche Intelligenz*, 29(2):193–201, 2015.

[93] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *European conference on computer vision*, pages 428–441. Springer, 2006.

[94] Cristian Sminchisescu and Bill Triggs. Kinematic jump processes for monocular 3d human tracking. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2003.

[95] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4293–4302, 2016.

[96] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[97] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 525–531. IEEE, 2001.

[98] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[99] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[100] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.

[101] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, 2005.

[102] Xiaoyu Wang, Tony X Han, and Shuicheng Yan. An hog-lbp human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 32–39. IEEE, 2009.

[103] Christoph G Keller, Markus Enzweiler, Marcus Rohrbach, David Fernández Llorca, Christoph Schnorr, and Dariu M Gavrila. The benefits of dense stereo for pedestrian detection. *IEEE transactions on intelligent transportation systems*, 12(4):1096–1106, 2011.

[104] Lu Xia, Chia-Chih Chen, and Jake K Aggarwal. Human detection using depth information by kinect. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 15–22. IEEE, 2011.

[105] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust rgb-d object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 681–687. IEEE, 2015.

[106] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[107] Shinji Kawatsuma, Mineo Fukushima, and Takashi Okada. Emergency response by robots to Fukushima-Daiichi accident: summary and lessons learned. *Industrial Robot: An International Journal*, 39(5):428–435, 2012.

[108] Kristopher Toussaint, Nicolas Pouliot, and Serge Montambault. Transmission line maintenance robots capable of crossing obstacles: State-of-the-art review and challenges ahead. *JFR*, 26(5):477–499, 2009.

[109] David Silver, J Andrew Bagnell, and Anthony Stentz. Learning from demonstration for autonomous navigation in complex unstructured terrain. *IJRR*, 29(12):1565–1592, 2010.

[110] Ayse Naz Erkan, Raia Hadsell, Pierre Sermanet, Jan Ben, Urs Muller, and Yann LeCun. Adaptive long range vision in unstructured terrain. In *IROS*, 2007.

[111] Christopher A Brooks and Karl Iagnemma. Vibration-based terrain classification for planetary exploration rovers. *TRO*, 21(6):1185–1191, 2005.

[112] Ayanna Howard and Homayoun Seraji. Vision-based terrain characterization and traversability assessment. *Journal of Robotic Systems*, 18(10):577–587, 2001.

[113] Sang-Ho Hyon. Compliant terrain adaptation for biped humanoids without measuring ground surface and contact forces. *TRO*, 25(1):171–178, 2009.

[114] Yoon-Gu Kim, Jeong-Hwan Kwak, Dae-Han Hong, In-Huck Kim, Dong-Hwan Shin, and Jinung An. Autonomous terrain adaptation and user-friendly tele-operation of wheel-track hybrid mobile robot. *International Journal of Precision Engineering and Manufacturing*, 13 (10):1781–1788, 2012.

[115] Michael Shneier, Tommy Chang, Tsai Hong, Will Shackleford, Roger Bostelman, and James S Albus. Learning traversability models for autonomous mobile vehicles. *AuRo*, 24 (1):69–86, 2008.

[116] Panagiotis Papadakis. Terrain traversability analysis methods for unmanned ground vehicles: A survey. *Engineering Applications of Artificial Intelligence*, 26(4):1373–1385, 2013.

[117] Maggie Wigness, John G Rogers, and Luis E Navarro-Serment. Robot navigation from human demonstration: learning control behaviors. In *ICRA*, 2018.

[118] Fernando L Garcia Bermudez, Ryan C Julian, Duncan W Haldane, Pieter Abbeel, and Ronald S Fearing. Performance analysis and terrain classification for a legged robot over rough terrain. In *IROS*, 2012.

[119] Dylan Williamson, Navinda Kottege, and Peyman Moghadam. Terrain characterisation and gait adaptation by a hexapod robot. In *Australasian Conference on Robotics and Automation (ACRA)*, 2016.

[120] Christian Weiss, Holger Frohlich, and Andreas Zell. Vibration-based terrain classification using support vector machines. In *IROS*, 2006.

[121] Michael Happold, Mark Ollis, and Nikolas Johnson. Enhancing supervised terrain classification with predictive unsupervised learning. In *RSS*, 2006.

[122] Thomas M Howard and Alonzo Kelly. Optimal rough terrain trajectory generation for wheeled mobile robots. *IJRR*, 26(2):141–166, 2007.

[123] Anelia Angelova, Larry Matthies, Daniel Helmick, Gabe Sibley, and Pietro Perona. Learning to predict slip for ground robots. In *ICRA*, 2006.

[124] Sebastian Thrun, Michael Montemerlo, and Andrei Aron. Probabilistic terrain analysis for high-speed desert driving. In *RSS*, 2006.

[125] Paul Filitchkin and Katie Byl. Feature-based terrain classification for littledog. In *IROS*, 2012.

[126] Stefan Laible, Yasir Niaz Khan, and Andreas Zell. Terrain classification with conditional random fields on fused 3D LIDAR and camera data. In *European Conference on Mobile Robots*, 2013.

[127] Mieczyslaw Gregory Bekker. *Introduction to terrain-vehicle systems*. University of Michigan Press, 1969.

[128] Chris Urmson, Charlie Ragusa, David Ray, Joshua Anhalt, Daniel Bartz, Tugrul Galatali, Alexander Gutierrez, Josh Johnston, Sam Harbaugh, Hiroki "Yu" Kato, et al. A robust approach to high-speed navigation for unrehearsed desert terrain. *JFR*, 23(8):467–508, 2006.

[129] Cheng Wang and Nancy F Glenn. Integrating LiDAR intensity and elevation data for terrain characterization in a forested area. *IEEE Geoscience and Remote Sensing Letters*, 6(3): 463–466, 2009.

[130] Karl Iagnemma, Shinwoo Kang, Hassan Shibly, and Steven Dubowsky. Online terrain parameter estimation for wheeled mobile robots with application to planetary rovers. *TRO*, 20(5):921–927, 2004.

[131] Lauro Ojeda, Johann Borenstein, Gary Witus, and Robert Karlsen. Terrain characterization and classification with a mobile robot. *JFR*, 23(2):103–122, 2006.

[132] Eric Trautmann and Laura Ray. Mobility characterization for autonomous mobile robots using machine learning. *AuRo*, 30(4):369–383, 2011.

[133] Roberto Manduchi, Andres Castano, Ashit Talukder, and Larry Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *AuRo*, 18(1):81–102, 2005.

[134] Lynne E Parker. Task-oriented multi-robot learning in behavior-based systems. In *IROS*, 1996.

[135] Lynne E Parker. Case study for life-long learning and adaptation in coopertive robot teams. In *Sensor Fusion and Decentralized Control in Robotic Systems II*, volume 3839, pages 92–102. International Society for Optics and Photonics, 1999.

[136] Jonathan Ko and Dieter Fox. GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models. *AuRo*, 27(1):75–90, 2009.

[137] Marc Deisenroth and Carl E Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In *ICML*, pages 465–472, 2011.

[138] Ian Lenz, Ross A Knepper, and Ashutosh Saxena. DeepMPC: Learning deep latent features for model predictive control. In *RSS*, 2015.

[139] Christian Plagemann, Sebastian Mischke, Sam Prentice, Kristian Kersting, Nicholas Roy, and Wolfram Burgard. Learning predictive terrain models for legged robot locomotion. In *IROS*, 2008.

[140] Sergey Levine and Pieter Abbeel. Learning neural network policies with guided policy search under unknown dynamics. In *NIPS*, 2014.

[141] Austin D Buchan, Duncan W Haldane, and Ronald S Fearing. Automatic identification of dynamic piecewise affine models for a running robot. In *IROS*, 2013.

[142] Gentaro Taga. A model of the neuro-musculo-skeletal system for human locomotion. *Biological Cybernetics*, 73(2):97–111, 1995.

[143] Hiroshi Kimura, Yasuhiro Fukuoka, and Hiroyuki Nakamura. Biologically inspired adaptive dynamic walking of the quadruped on irregular terrain. In *Robotics Research*, pages 329–336. 2000.

[144] Hiroshi Kimura, Seiichi Akiyama, and Kazuaki Sakurama. Realization of dynamic walking and running of the quadruped using neural oscillator. *AuRo*, 7(3):247–258, 1999.

[145] Andrew Lookingbill, John Rogers, David Lieb, J Curry, and Sebastian Thrun. Reverse optical flow for self-supervised adaptive autonomous robot navigation. *IJCV*, 74(3):287–302, 2007.

[146] Samir Nabulsi, M Armada, and Héctor Montes. Multiple terrain adaptation approach using ultrasonic sensors for legged robots. In *Climbing and Walking Robots*. 2006.

[147] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *RAM*, 4(1):23–33, 1997.

[148] Zheng Zhang, Yong Xu, Jian Yang, Xuelong Li, and David Zhang. A survey of sparse representation: algorithms and applications. *IEEE Access*, 3:490–530, 2015.

[149] Sylvain Calinon and Aude Billard. Incremental learning of gestures by imitation in a humanoid robot. In *HRI*, 2007.

[150] Nathan Ratliff, J Andrew Bagnell, and Siddhartha S Srinivasa. Imitation learning for locomotion and manipulation. In *Humanoids*, 2007.

[151] Feiping Nie, Heng Huang, Xiao Cai, and Chris H Ding. Efficient and robust feature selection via joint 2, 1-norms minimization. In *NIPS*, 2010.

[152] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face Description with Local Binary Patterns: Application to Face Recognition. *PAMI*, (12):2037–2041, 2006.

[153] Péter Fankhauser, Michael Bloesch, Christian Gehring, Marco Hutter, and Roland Siegwart. Robot-centric elevation mapping with uncertainty estimates. In *Mobile Service Robotics*, pages 433–440. 2014.

[154] Fenglu Ge, Wayne Moore, and Michael Antolovich. Learning from demonstration using GMM, CHMM and DHMM: A comparison. In *Australasian Joint Conference on Artificial Intelligence*, 2015.

[155] Shingo Shimoda, Yoji Kuroda, and Karl Iagnemma. Potential field navigation of high speed unmanned ground vehicles on uneven terrain. In *ICRA*, 2005.

[156] David Lattanzi and Gregory Miller. Review of Robotic Infrastructure Inspection Systems. *Journal of Infrastructure Systems*, 23(3):04017004, 2017.

[157] Martin J Schuster, Sebastian G Brunner, Kristin Bussmann, Stefan Büttner, Andreas Dömel, Matthias Hellerer, Hannah Lehner, Peter Lehner, Oliver Porges, Josef Reill, et al. Towards Autonomous Planetary Exploration. *JINT*, 2019.

[158] Hao Tien Lewis Chiang, Baisravan HomChaudhuri, Lee Smith, and Lydia Tapia. Safety, challenges, and performance of motion planners in dynamic environments. In *Robotics Research*, pages 793–808. Springer, 2020.

[159] Guillaume Sartoretti, Samuel Shaw, Katie Lam, Naixin Fan, Matthew Travers, and Howie Choset. Central pattern generator with inertial feedback for stable locomotion and climbing in unstructured terrain. In *International Conference on Robotics and Automation*, 2018.

[160] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *Transactions on Robotics and Automation*, 1996.

[161] Grady Williams, Paul Drews, Brian Goldfain, James M Rehg, and Evangelos A Theodorou. Aggressive driving with model predictive path integral control. In *International Conference on Robotics and Automation*, 2016.

[162] Lazaros Moysis, Eleftherios Petavratzis, Christos Volos, Hector Nistazakis, and Ioannis Stouboulos. A Chaotic Path Planning Generator Based on Logistic Map and Modulo Tactics. *RAS*, 2020.

[163] Gregory Kahn, Pieter Abbeel, and Sergey Levine. LaND: Learning to Navigate from Disengagements. *RAL*, 2021.

[164] Gregory Kahn, Pieter Abbeel, and Sergey Levine. BADGR: an autonomous self-supervised learning-based navigation system. *Robotics and Automation Letters*, 6(2):1312–1319, 2021.

[165] Gregory Kahn, Adam Villaflor, Bosen Ding, Pieter Abbeel, and Sergey Levine. Self-supervised Deep Reinforcement Learning with Generalized Computation Graphs for Robot Navigation. In *ICRA*, 2018.

[166] Seung-Ho Han, Ho-Jin Choi, Philipp Benz, and Jorge Loaiciga. Sensor-Based Mobile Robot Navigation via Deep Reinforcement Learning. In *ICBDSC*, 2018.

[167] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid Motor Adaptation for Legged Robots. *RSS*, 2021.

[168] Bo Liu, Xuesu Xiao, and Peter Stone. A Lifelong Learning Approach to Mobile Robot Navigation. *RAL*, 2021.

[169] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual Learning through Synaptic Intelligence. In *ICML*, 2017.

[170] Gregory Kahn, Adam Villaflor, Pieter Abbeel, and Sergey Levine. Composable Action-Conditioned Predictors: Flexible Off-policy Learning for Robot Navigation. In *CoRL*, 2018.

[171] Jyun-Yu Jhang, Cheng-Jian Lin, Chin-Teng Lin, and Kuu-Young Young. Navigation Control of Mobile Robots Using an Interval Type-2 Fuzzy Controller Based on Dynamic-group Particle Swarm Optimization. *IJCAS*, 2018.

[172] Aman Sinha, Matthew O'Kelly, Russ Tedrake, and John C Duchi. Neural Bridge Sampling for Evaluating Safety-Critical Autonomous Systems. *NIPS*, 2020.

[173] Aman Sinha, Matthew O'Kelly, Hongrui Zheng, Rahul Mangharam, John Duchi, and Russ Tedrake. Formulazero: Distributionally Robust Online Adaptation via Offline Population Synthesis. In *ICML*, 2020.

[174] Alessandro Saffiotti. The uses of Fuzzy Logic in Autonomous Robot Navigation. *Soft computing*, 1997.

[175] Meng Wang and James NK Liu. Fuzzy Logic-Based Real-Time Robot Navigation in Unknown Environment with Dead Ends. *RAS*, 2008.

[176] A Johansson, T Haraldson, R Omar, S Kiliaridis, and GE Carlsson. A System for Assessing the Severity and Progression of Occlusal Tooth Wear. *Journal of oral rehabilitation*, 1993.

[177] L Rabiner, R Crochiere, and J Allen. FIR System Modeling and Identification in the Presence of Noise and with Band-Limited Inputs. *ICASSP*, 1978.

[178] Yolanda Bolea, Antoni Grau, and Alberto Sanfeliu. Non-speech Sound Feature Extraction Based on Model Identification for Robot Navigation. In *CIARP*, 2003.

[179] Dwi Pebrianti, Yong Hooi Hao, Nur Aisyah Syafinaz Suarin, Luhur Bayuaji, Zulkifli Musa, Mohammad Syafrullah, and Indra Riyanto. Motion Tracker Based Wheeled Mobile Robot System Identification and Controller Design. In *SympoSIMM*. 2018.

[180] Jiongmin Yong and Xun Yu Zhou. *Stochastic Controls: Hamiltonian Systems and HJB Equations*. Springer Science & Business Media, 1999.

[181] Jur Van Den Berg, Sachin Patil, and Ron Alterovitz. Motion Planning under Uncertainty using differential Dynamic Programming in Belief Space. In *Robotics Research*. 2017.

[182] Hao-jie Zhang, Jian-wei Gong, Yan Jiang, Guang-ming Xiong, and Hui-yan Chen. An Iterative Linear Quadratic Regulator based Trajectory Tracking Controller for Wheeled Mobile Robot. *Journal of Zhejiang University*, 2012.

[183] Thomas M Howard, Colin J Green, and Alonzo Kelly. Receding Horizon Model-Predictive Control for Mobile Robot Navigation of Intricate Paths. In *FSR*, 2010.

[184] Osama Abdul Hafez, Guillermo Duenas Arana, and Matthew Spenko. Integrity Risk-Based Model Predictive Control for Mobile Robots. In *ICRA*, 2019.

[185] Adnan Tahirovic and Gianantonio Magnani. General Framework for Mobile Robot Navigation using Passivity-based MPC. *TACON*, 2010.

[186] Bernard O Koopman. Hamiltonian Systems and Transformation in Hilbert Space. *PNAS*, 1931.

[187] Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. Generalizing Koopman theory to allow for inputs and control. *Journal on Applied Dynamical Systems*, 17(1):909–930, 2018.

[188] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition. *J. Nonlinear Sci.*, 2015.

[189] Christopher G Atkeson and Stefan Schaal. Robot Learning from Demonstration. In *ICML*, 1997.

[190] David Silver, J Andrew Bagnell, and Anthony Stentz. Applied Imitation Learning for Autonomous Navigation in Complex Natural Terrain. In *FSR*, 2010.

[191] Fatemeh Fathinezhad, Vali Derhami, and Mehdi Rezaeian. Supervised Fuzzy Reinforcement Learning for Robot Navigation. *Applied Soft Computing*, 2016.

[192] Zizhao Wang, Xuesu Xiao, Bo Liu, Garrett Warnell, and Peter Stone. Appli: Adaptive Planner Parameter Learning from Interventions. In *ICRA*, 2021.

[193] MG Harinarayanan Nampoothiri, B Vinayakumar, Youhan Sunny, and Rahul Antony. Recent Developments in Terrain Identification, Classification, Parameter Estimation for the Navigation of Autonomous Robots. *SN Applied Sciences*, 2021.

[194] Thomas Duriez, Steven L Brunton, and Bernd R Noack. *Machine Learning Control-Taming Nonlinear Dynamics and Turbulence*. Springer, 2017.

[195] Steven L Brunton and J Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

[196] Peter J Schmid. Dynamic Mode Decomposition of Numerical and Experimental Data. *Journal of fluid mechanics*, 2010.

[197] Jonathan H Tu. *Dynamic Mode Decomposition: Theory and Applications*. PhD thesis, Princeton University, 2013.

[198] Giorgos Mamakoukas, Maria Castano, Xiaobo Tan, and Todd Murphey. Local Koopman Operators for Data-Driven Control of Robotic Systems. In *RSS*, 2019.

[199] Hao Wang and Noboru Noguchi. Real-time States Estimation of a Farm Tractor using Dynamic Mode Decomposition. *GPS Solutions*, 2021.

[200] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic Mode Decomposition: Data-driven Modeling of Complex Systems*. SIAM, 2016.

[201] Camilo Cáceres, João Mauricio Rosário, and Dario Amaya. Approach of Kinematic Control for a Non-Holonomic Wheeled Robot using Artificial Neural Networks and Genetic Algorithms. In *IWOBI*, 2017.

[202] Daniel R Ramírez, Daniel Limón, J Gomez-Ortega, and Eduardo F Camacho. Nonlinear MBPC for mobile robot navigation using genetic algorithms. In *ICRA*, 1999.

[203] Morgan T Gillespie, Charles M Best, Eric C Townsend, David Wingate, and Marc D Killpack. Learning Nonlinear Dynamic Models of Soft Robots for Model Predictive Control with Neural Networks. In *RoboSoft*, 2018.

[204] An iterative lqr controller for off-road and on-road vehicles using a neural network dynamics model.

[205] Mohammed Alharbi and Hassan A Karimi. A Global Path Planner for Safe Navigation of Autonomous Vehicles in Uncertain Environments. *Sensors*, 2020.

[206] Adrià Garriga-Alonso, Carl Edward Rasmussen, and Laurence Aitchison. Deep Convolutional Networks as Shallow Gaussian Processes. *ICLR*, 2019.

[207] Zhilu Zhang and Mert R Sabuncu. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. In *NIPS*, 2018.

[208] Lawrence Chan, Dylan Hadfield-Menell, Siddhartha Srinivasa, and Anca Dragan. The Assistive Multi-Armed Bandit. In *HRI*, 2019.

[209] Krishnakumar Balasubramanian and Saeed Ghadimi. Zeroth-Order Nonconvex Stochastic Optimization: Handling Constraints, High Dimensionality, and Saddle Points. *FCoM*, 2021.

[210] Gerasimos G Rigatos. Extended Kalman and Particle Filtering for Sensor Fusion in Motion Control of Mobile Robots. *IMACS*, 2010.

[211] Anirban Mazumdar, Meng Yee Chuah, Michael S Triantafyllou, and H Harry Asada. Design for Precision Multi-Directional Maneuverability: Egg-Shaped Underwater Robots for Infrastructure Inspection. In *International Conference on Robotics and Automation*, 2014.

[212] Keiji Nagatani, Seiga Kiribayashi, Yoshito Okada, Kazuki Otake, Kazuya Yoshida, Satoshi Tadokoro, Takeshi Nishimura, Tomoaki Yoshida, Eiji Koyanagi, Mineo Fukushima, et al. Emergency Response to the Nuclear Accident at the Fukushima Daiichi Nuclear Power Plants using Mobile Rescue Robots. *JFR*, 2013.

[213] Roland Siegwart, Marco Hutter, Philipp Oettershagen, Michael Burri, Igor Gilitschenski, Enric Galceran, and Juan Nieto. Legged and Flying Robots for Disaster Response. In *World Engineering Conference and Convention*, 2015.

[214] Antigoni Tsiami, Petros Koutras, Niki Efthymiou, Panagiotis Paraskevas Filntisis, Gerasimos Potamianos, and Petros Maragos. Multi3: Multi-Sensory Perception System for Multi-Modal Child Interaction with Multiple Robots. In *International Conference on Robotics and Automation*, 2018.

[215] Amir Salimi Lafmejani and Spring Berman. Nonlinear MPC for Collision-free and Deadlock-free Navigation of Multiple Non-holonomic Mobile Robots. *Robotics and Autonomous Systems*, 2021.

[216] N Wagener, C Cheng, J Sacks, and B Boots. An Online Learning Approach to Model Predictive Control. In *Robotics: Science and Systems*, 2019.

[217] Zifan Xu, Gauraang Dhamankar, Anirudh Nair, Xuesu Xiao, Garrett Warnell, Bo Liu, Zizhao Wang, and Peter Stone. APPLR: Adaptive Planner Parameter Learning from Reinforcement. In *International Conference on Robotics and Automation*, 2021.

[218] Yifeng Zhu, Devin Schwab, and Manuela Veloso. Learning Primitive Skills for Mobile Robots. In *International Conference on Robotics and Automation*, 2019.

[219] Beomjoon Kim and Joelle Pineau. Socially Adaptive Path Planning in Human Environments using Inverse Reinforcement Learning. *International Journal of Social Robotics*, 2016.

[220] Daniel Gusland, Børge Torvik, Erlend Finden, Fredrik Gulbrandsen, and Ragnar Smestad. Imaging radar for navigation and surveillance on an autonomous unmanned ground vehicle capable of detecting obstacles obscured by vegetation. In *Radar Conference*, 2019.

[221] Raja Chatila, Erwan Renaudo, Mihai Andries, Ricardo-Omar Chavez-Garcia, Pierre Luce-Vayrac, Raphaël Gottstein, Rachid Alami, Aurélie Clodic, Sandra Devin, Benoît Girard, et al. Toward Self-Aware Robots. *Frontiers in Robotics and AI*, 2018.

[222] Yoshiteru Ishida. A Note on Continuous Self-Identification as Self-Awareness: An Example of Robot Navigation. *Procedia Computer Science*, 2015.

[223] Siwei Zhang, Robert Pöhlmann, Thomas Wiedemann, Armin Dammann, Henk Wymeersch, and Peter Adam Hoeher. Self-Aware Swarm Navigation in Autonomous Exploration Missions. *Proceedings of the IEEE*, 2020.

[224] Sarah Khan and Richard Chang. Anatomy of the Vestibular System: A Review. *NeuroRehabilitation*, 2013.

[225] Roland S Johansson. Sensory Input and Control of Grip. In *Novartis Foundation Symposium 218-Sensory Guidance of Movement: Sensory Guidance of Movement: Novartis Foundation Symposium 218*, 2007.

[226] Mohamed Slim Masmoudi, Najla Krichen, Mohamed Masmoudi, and Nabil Derbel. Fuzzy Logic Controllers Design for Omnidirectional Mobile Robot Navigation. *Applied soft computing*, 2016.

[227] Homayoun Seraji and Ayanna Howard. Behavior-Based Robot Navigation on Challenging Terrain: A Fuzzy Logic Approach. *IEEE Transactions on Robotics and Automation*, 2002.

[228] Ulrich Nehmzow, Otar Akanyeti, Christoph Weinrich, Theocharis Kyriacou, and Stephen A Billings. Robot Programming by Demonstration through System Identification. In *International Conference on Intelligent Robots and Systems*, 2007.

[229] Umar Iqbal, Aboelmagd Noureldin, Jacques Georgy, and Michael J Korenberg. Application of System Identification Techniques for Integrated Navigation. In *International Conference on Communications, Signal Processing, and their Applications*, 2021.

[230] Mostafa Nazari, Javad Amiryan, and Eslam Nazemi. Improvement of Robot Navigation using Fuzzy Method. In *Joint Conference of AI & Robotics and 5th RoboCup Iran Open International Symposium*, 2013.

[231] Yuichiro Aoyama, George Boutselis, Akash Patel, and Evangelos A Theodorou. Constrained Differential Dynamic Programming Revisited. In *International Conference on Robotics and Automation*, 2021.

[232] Shatil Rahman and Steven L Waslander. Uncertainty-Constrained Differential Dynamic Programming in Belief Space for Vision Based Robots. *Robotics and Automation Letters*, 2021.

[233] David Fridovich-Keil, Ellis Ratner, Lasse Peters, Anca D Dragan, and Claire J Tomlin. Efficient Iterative Linear-Quadratic Approximations for Nonlinear Multi-Player General-Sum Differential Games. In *International Conference on Robotics and Automation*, 2020.

[234] Joanne Truong, Sonia Chernova, and Dhruv Batra. Bi-directional Domain Adaptation for Sim2Real Transfer of Embodied Navigation Agents. *Robotics and Automation Letters*, 2021.

[235] Chris J Ostafew, Angela P Schoellig, Timothy D Barfoot, and Jack Collier. Learning-Based Nonlinear Model Predictive Control to Improve Vision-based Mobile Robot Path Tracking. *JFR*, 2016.

[236] Qidan Zhu, Yu Han, Chengtao Cai, and Yao Xiao. Robust Optimal Navigation Using Nonlinear Model Predictive Control Method Combined With Recurrent Fuzzy Neural Network. *Mathematical Problems in Engineering*, 2018.

[237] M Rana, Mustafa Mukadam, Seyed Reza Ahmadzadeh, Sonia Chernova, and Byron Boots. Towards Robust Skill Generalization: Unifying Learning from Demonstration and Motion Planning. In *Intelligent robots and systems*, 2018.

[238] Boris Sofman, Ellie Lin, J Andrew Bagnell, John Cole, Nicolas Vandapel, and Anthony Stentz. Improving Robot Navigation through Self-supervised Online Learning. *JFR*, 2006.

[239] Zhe Wu, Anh Tran, David Rincon, and Panagiotis D Christofides. Machine Learning-based Predictive Control of Nonlinear Processes. Part I: Theory. *AIChE Journal*, 2019.

[240] Shiyu Yang, Man Pun Wan, Wanyu Chen, Bing Feng Ng, and Swapnil Dubey. Model Predictive Control with Adaptive Machine-Learning-Based Model for Building Energy Efficiency and Comfort Optimization. *Applied Energy*, 2020.

[241] Giuseppe L'Erario, Luca Fiorio, Gabriele Nava, Fabio Bergonti, Hosameldin Awadalla Omer Mohamed, Emilio Benenati, Silvio Traversaro, and Daniele Pucci. Modeling, Identification and Control of Model Jet Engines for Jet Powered Robotics. *RAL*, 2020.

[242] Cosimo Della Santina, Matteo Bianchi, Giorgio Grioli, Franco Angelini, Manuel Catalano, Manolo Garabini, and Antonio Bicchi. Controlling Soft Robots: Balancing Feedback and Feedforward Elements. *Robotics Automation Magazine*, 2017.

[243] D Santosh, Supreeth Achar, and CV Jawahar. Autonomous Image-based Exploration for Mobile Robot Navigation. In *International Conference on Robotics and Automation*, 2008.

[244] KI Kiy. Segmentation and Detection of Contrast Objects and their Application in Robot Navigation. *Pattern Recognition and Image Analysis*, 2015.

[245] Alejandra Carolina Hernández, Clara Gómez, Jonathan Crespo, and Ramón Barber. Object Detection Applied to Indoor Environments for Mobile Robot Navigation. *Sensors*, 2016.

[246] Carlos Astua, Ramon Barber, Jonathan Crespo, and Alberto Jardon. Object detection techniques applied on mobile robot semantic navigation. *Sensors*, 2014.

[247] Christopher Urmson, Joshua Anhalt, Michael Clark, Tugrul Galatali, Juan Pablo Gonzalez, Jay Gowdy, Alexander Gutierrez, Sam Harbaugh, Matthew Johnson-Roberson, Hiroki Kato, et al. High speed navigation of unrehearsed terrain: Red team technology for grand challenge 2004. *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-04-37*, 2004.

[248] Naim Sidek and Nilanjan Sarkar. Exploiting wheel slips of mobile robots to improve navigation performance. *Advanced Robotics*, 27(8):627–639, 2013.

[249] Debangshu Sadhukhan, Carl Moore, and Emmanuel Collins. Terrain estimation using internal sensors. In *International Conference on Robotics and Applications*, 2004.

[250] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[251] Peng Jiang, Philip Osteen, Maggie Wigness, and Srikanth Saripalli. RELLIS-3D Dataset: Data, Benchmarks and Analysis.

[252] Maggie Wigness, Sungmin Eum, John G Rogers, David Han, and Heesung Kwon. A RUGD dataset for autonomous navigation and visual perception in unstructured outdoor environments. In *International Conference on Intelligent Robots and Systems*, 2019.

[253] Jim Hollinger, Brett Kutscher, and Ryan Close. Fusion of Lidar and Radar for Detection of Partially Obscured Objects. In *Unmanned Systems Technology XVII*, 2015.

[254] Zhengping Ji, Matthew D Luciw, Juyang Weng, Shuqing Zeng, and Varsha Sadekar. A Biologically-Motivated Developmental System for Perceptual Awareness in Vehicle-Based Robots. In *International Conference on Epigenetic Robotics*, 2007.

[255] Sherwin A Guirnaldo, Keigo Watanabe, and Kiyotaka Izumi. Enhancing the Awareness of Decentralized Cooperative Mobile Robots through Active Perceptual Anchoring. *International Journal of Control, Automation, and Systems*, 2004.

[256] Jun Jin, Nhat M Nguyen, Nazmus Sakib, Daniel Graves, Hengshuai Yao, and Martin Jagersand. Mapless navigation among dynamics with social-safety-awareness: a reinforcement learning approach from 2d laser scans. In *International Conference on Robotics and Automation*, 2020.

[257] Brandon Amos, Laurent Dinh, Serkan Cabi, Thomas Rothörl, Sergio Gómez Colmenarejo, Alistair Muldal, Tom Erez, Yuval Tassa, Nando de Freitas, and Misha Denil. Learning awareness models. In *International Conference on Learning Representations*, 2018.

[258] Xilong Liu, Zhiqiang Cao, Yingying Yu, Guangli Ren, Junzhi Yu, and Min Tan. Robot Navigation Based on Situational Awareness. *Transactions on Cognitive and Developmental Systems*, 2021.

[259] Naman Patel, Anna Choromanska, Prashanth Krishnamurthy, and Farshad Khorrami. A Deep Learning Gated Architecture for UGV Navigation Robust to Sensor Failures. *Robotics and Autonomous Systems*, 2019.

[260] Zachary Manchester and Scott Kuindersma. Robust Direct Trajectory Optimization using Approximate Invariant Funnels. *Autonomous Robots*, 2019.

[261] Andreas Fischer. A special Newton-type optimization method. *Optimization*, 24(3-4): 269–284, 1992.

[262] Péter Fankhauser, Michael Bloesch, and Marco Hutter. Probabilistic Terrain Mapping for Mobile Robots with Uncertain Localization. *Robotics and Automation Letters*, 2018. doi: 10.1109/LRA.2018.2849506.

[263] Al Abdulla Hmoudi. Robotics for Disaster Warning and Response in the UAE. *Journal of Environmental Science and Engineering*, 9:215–221, 2020.

[264] L Bruzzone and Giuseppe Quaglia. Locomotion systems for ground mobile robots in unstructured environments. *Mechanical Sciences*, 3(2):49–62, 2012.

[265] EA Devjanin, VS Gurfinkel, EV Gurfinkel, VA Kartashev, AV Lensky, A Yu Shneider, and LG Shtilman. The six-legged walking robot capable of terrain adaptation. *Mechanism and Machine Theory*, 18(4):257–260, 1983.

[266] Edmond M DuPont, Carl A Moore, and Rodney G Roberts. Terrain classification for mobile robots traveling at various speeds: An eigenspace manifold approach. In *International Conference on Robotics and Automation*, 2008.

[267] Robert Hudjakov and Mart Tamre. Aerial imagery terrain classification for long-range autonomous navigation. In *International Symposium on Optomechatronic Technologies*, 2009.

[268] Thierry Peynot, Sin Ting Lui, Rowan McAllister, Robert Fitch, and Salah Sukkarieh. Learned stochastic mobility prediction for planning with control uncertainty on unstructured terrain. *Journal of Field Robotics*, 31(6):969–995, 2014.

[269] Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan Schaal. Learning and generalization of motor skills by learning from demonstration. In *International Conference on Robotics and Automation*, 2009.

[270] Carlos David Braga Borges, Antonio Márcio Albuquerque Almeida, Iális Cavalcante Paula Junior, and Jarbas Joaci de Mesquita Sá Junior. A strategy and evaluation method for ground global path planning based on aerial images. *Expert Systems With Applications*, 137: 232–252, 2019.

[271] Russell Knight, Forest Fisher, Tara Estlin, Barbara Engelhardt, and Steve Chien. Balancing deliberation and reaction, planning and execution for space robotic applications. In *International Conference on Intelligent Robots and Systems*, 2001.

[272] AR Jumikis. Introduction to Terrain-Vehicle Systems. *Soil Science*, 110(1):77, 1970.

[273] Matthew Spenko, Yoji Kuroda, Steven Dubowsky, and Karl Iagnemma. Hazard avoidance for high-speed mobile robots in rough terrain. *JFR*, 23(5):311–331, 2006.

[274] Peng Jiang, Philip Osteen, Maggie Wigness, and Srikanth Saripalli. RELLIS-3D Dataset: Data, Benchmarks and Analysis, 2020.

[275] Bin He, Shuai Wang, and Yongjia Liu. Underactuated robotics: a review. *International Journal of Robotics Research*, 16(4), 2019.

[276] Lynne E Parker. L-ALLIANCE: Task-oriented multi-robot learning in behavior-based systems. *Advanced Robotics*, 11(4):305–322, 1996.

[277] Sebastian Thrun. Lifelong learning algorithms. In *Learning to Learn*. Springer, 1998.

[278] Seung Joon Yi, Byoung Tak Zhang, and Daniel Lee. Online learning of uneven terrain for humanoid bipedal walking. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, volume 24, 2010.

[279] Alec Koppel, Jonathan Fink, Garrett Warnell, Ethan Stump, and Alejandro Ribeiro. Online learning for characterizing unknown environments in ground robotic vehicle models. In *International Conference on Intelligent Robots and Systems*, 2016.

[280] Karime Pereida and Angela P Schoellig. Adaptive model predictive control for high-accuracy trajectory tracking in changing conditions. In *International Conference on Intelligent Robots and Systems*, 2018.

[281] Seongchan Jeong and Dongkyoung Chwa. Coupled multiple sliding-mode control for robust trajectory tracking of hovercraft with external disturbances. *Transactions on Industrial Electronics*, 65(5):4103–4113, 2017.

[282] Aytaç Altan and Rıfat Hacıoğlu. Model predictive control of three-axis gimbal system mounted on UAV for real-time target tracking under external disturbances. *Mechanical Systems and Signal Processing*, 138:106548, 2020.

[283] Haiping Lu, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. A survey of multilinear subspace learning for tensor data. *Pattern Recognition*, 44(7):1540–1551, 2011.

[284] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *International Conference on Intelligent Robots and Systems*, 2012.

[285] Yaser Alothman and Dongbing Gu. Quadrotor transporting cable-suspended load using iterative linear quadratic regulator (iLQR) optimal control. In *Computer Science and Electronic Engineering*, 2016.

[286] Ian Abraham and Todd D Murphey. Active learning of dynamics for data-driven control using Koopman operators. *Transactions on Robotics*, 35(5):1071–1083, 2019.

[287] Tixiao Shan and Brendan Englot. LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain. In *International Conference on Intelligent Robots and Systems*, 2018.

[288] Yan Wu, Ruohan Wang, Luis F D'Haro, Rafael E Banchs, and Keng Peng Tee. Multi-modal robot apprenticeship: Imitation learning using linearly decayed dmp+ in a human-robot dialogue system. In *International Conference on Intelligent Robots and Systems*, 2018.

[289] Zhuli Ren, Liguan Wang, and Lin Bi. Robust GICP-based 3D LiDAR SLAM for underground mining environment. *Sensors*, 19(13):2915, 2019.

[290] Yang Song, Mingyang Guan, Wee Peng Tay, Choi Look Law, and Changyun Wen. UWB/LiDAR fusion for cooperative range-only SLAM. In *International Conference on Robotics and Automation*, 2019.

[291] Jitong Zhang, Mingrong Ren, Pu Wang, Juan Meng, and Yuman Mu. Indoor Localization Based on VIO System and Three-Dimensional Map Matching. *Sensors*, 20(10):2790, 2020.

[292] Shinsuk Park, Yoojin Oh, and Daehie Hong. Disaster Response and Recovery from the Perspective of Robotics. *International Journal of Precision Engineering and Manufacturing*, 2017.

[293] Max Schwarz, Tobias Rodehutskors, David Droeschel, Marius Beul, Michael Schreiber, Nikita Araslanov, Ivan Ivanov, Christian Lenz, Jan Razlaw, Sebastian Schüller, et al. NimbRo Rescue: Solving Disaster-Response Tasks with the Mobile Manipulation Robot Momaro. *Journal of Field Robotics*, 2017.

[294] Helge-Björn Kuntze, Christian W Frey, Igor Tchouchenkov, Barbara Staehle, Erich Rome, Kai Pfeiffer, Andreas Wenzel, and Jürgen Wöllenstein. Seneka-Sensor Network with Mobile Robots for Disaster Management. In *Conference on Technologies for Homeland Security*, 2012.

[295] Shankar P Bhattacharyya, Aniruddha Datta, and Lee H Keel. *Linear Control Theory: Structure, Robustness, and Optimization*. CRC press, 2018.

[296] Masahiko Haruno, Daniel M Wolpert, and Mitsuo Kawato. Mosaic Model for Sensorimotor Learning and Control. *Neural computation*, 2001.

[297] Luis Felipe Posada, Krishna Kumar Narayanan, Frank Hoffmann, and Torsten Bertram. Floor segmentation of omnidirectional images for mobile robot visual navigation. In *International Conference on Intelligent Robots and Systems*, 2010.

[298] Stuart Marcovitch, Sophie Jacques, Janet J Boseovski, and Philip David Zelazo. Self-Reflection and the Cognitive Control of Behavior: Implications for Learning. *Mind, Brain, and Education*, 2008.

[299] Roger Smith. Self-Reflection and the Self. In *Rewriting the Self*. 2002.

[300] Monika Ardelt and Sabine Grunwald. The Importance of Self-Reflection and Awareness for Human Development in Hard Times. *Research in Human Development*, 2018.

[301] Fabian Rubilar, María-José Escobar, and Tomás Arredondo. Bio-Inspired Architecture for a Reactive-Deliberative Robot Controller. In *International Joint Conference on Neural Networks*, 2014.

[302] Juan Cristobal Zagal and Hod Lipson. Self-Reflection in Evolutionary Robotics: Resilient Adaptation with a Minimum of Physical Exploration. In *Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, 2009.

[303] Abdulrahman Altahhan. Self-Reflective Deep Reinforcement Learning. In *International Joint Conference on Neural Networks*, 2016.

[304] Hao Zhang, Christopher Reardon, Fei Han, and Lynne E Parker. Srac: Self-reflective risk-aware artificial cognitive models for robot response to human activities. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 3301–3308. IEEE, 2016.

[305] Ali Demir and Volkan Sezer. Motion Planning and Control with Randomized Payloads Using Deep Reinforcement Learning. In *International Conference on Robotic Computing*, 2019.

[306] Elliot W Hawkes, Carmel Majidi, and Michael T Tolley. Hard Questions for Soft Robotics. *Science robotics*, 2021.

[307] Stephan Rabanser, Oleksandr Shchur, and Stephan Günnemann. Introduction to Tensor Decompositions and their Applications in Machine Learning. 2017.

[308] Liqun Qi. Transposes, L-Eigenvalues and Invariants of Third Order Tensors. 2017.

[309] H Neudecker. A note on Kronecker Matrix Products and Matrix Equation Systems. *SIAM Journal on Applied Mathematics*, 17(3):603–606, 1969.

[310] Pengfei Chen, Guangyong Chen, and Shengyu Zhang. Log Hyperbolic Cosine Loss Improves Variational Auto-Encoder. 2018.

[311] Xing Xu, Jie Li, Yang Yang, and Fumin Shen. Towards Effective Intrusion Detection Using Log-cosh Conditional Variational AutoEncoder. *Internet of Things Journal*, 2020.

[312] Zhening Li, Yuji Nakatsukasa, Tasuku Soma, and André Uschmajew. On Orthogonal Tensors and Best Rank-One Approximation Ratio. *SIAM Journal on Matrix Analysis and Applications*, 2018.

[313] Mo Shan, Yingcai Bi, Hailong Qin, Jiaxin Li, Zhi Gao, Feng Lin, and Ben M Chen. A Brief Survey of Visual Odometry for Micro Aerial Vehicles. In *Annual Conference of the Industrial Electronics Society*, 2016.

[314] Anita S Gangal, PK Kalra, and DS Chauhan. Performance Evaluation of Complex Valued Neural Networks using Various Error Functions. *Enformatika*, 2007.

[315] Sahand Negahban and Martin J Wainwright. Restricted Strong Convexity and Weighted Matrix Completion: Optimal Bounds with Noise. *The Journal of Machine Learning Research*, 2012.

[316] Steven G. Johnson. The NLopt Nonlinear-Optimization Package. 2011.

[317] Christophe Grand, Faïz Benamar, Frédéric Plumet, and Philippe Bidaud. Stability and Traction Optimization of a Reconfigurable Wheel-Legged Robot. *The International Journal of Robotics Research*, 2004.

[318] Sriram Siva, Zachary Nahman, and Hao Zhang. Voxel-Based Representation Learning for Place Recognition Based on 3D Point Clouds. In *International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[319] Sriram Siva and Hao Zhang. Metacognitive Reasoning of Perceptual Inconsistency for Illusion Detection. In *Robotics: Science and Systems (RSS), Workshop Paper*, 2018.

[320] Nathaniel Boyless, Jiayi Liu, Sriram Siva, and Hao Zhang. Spherical Image Based Registration and Self-Localization for Onsite and Offsite Viewing. 2022. US Patent 11,418,716.

[321] Sriram Siva and Hao Zhang. Robot Perceptual Adaptation to Environment Changes for Long-Term Human Teammate Following. *International Journal of Robotics Research (IJRR)*, 41(7):706–720, 2021.

[322] Sriram Siva, Maggie Wigness, John Rogers, Long Quang, and Hao Zhang. Enhancing Consistent Ground Maneuverability by Robot Adaptation to Complex Off-Road Terrains. In Association for the Advancement of Artificial Intelligence *(AAAI)* SSS, 2021.

[323] Sriram Siva, Maggie Wigness, John G Rogers, Long Quang, and Hao Zhang. Self-Reflective Terrain-Aware Robot Adaptation for Consistent Off-Road Ground Navigation. *Submitted - International Journal of Robotics Research*, 2022.

[324] Sriram Siva, Maggie Wigness, John Rogers, Long Quang, and Hao Zhang. Self-Aware Perceptual Adaptation for Robust Ground Navigation in Unstructured Off-Road Environments. In *ArXiv Preprint*, 2022.

APPENDIX
COPYRIGHT PERMISSION

This dissertation consists of reproduced versions of papers that had co-authors and accepted by conferences. This appendix includes the publishers' reuse policy and permission to reuse works from co-authors.

A.1 IEEE, RSS, CoRL and Sage Reuse Policies

Chapters in this dissertation have been accepted to conferences administered by four organizations, IEEE, RSS, CoRL and Sage. Their reuse policies are listed as follows:

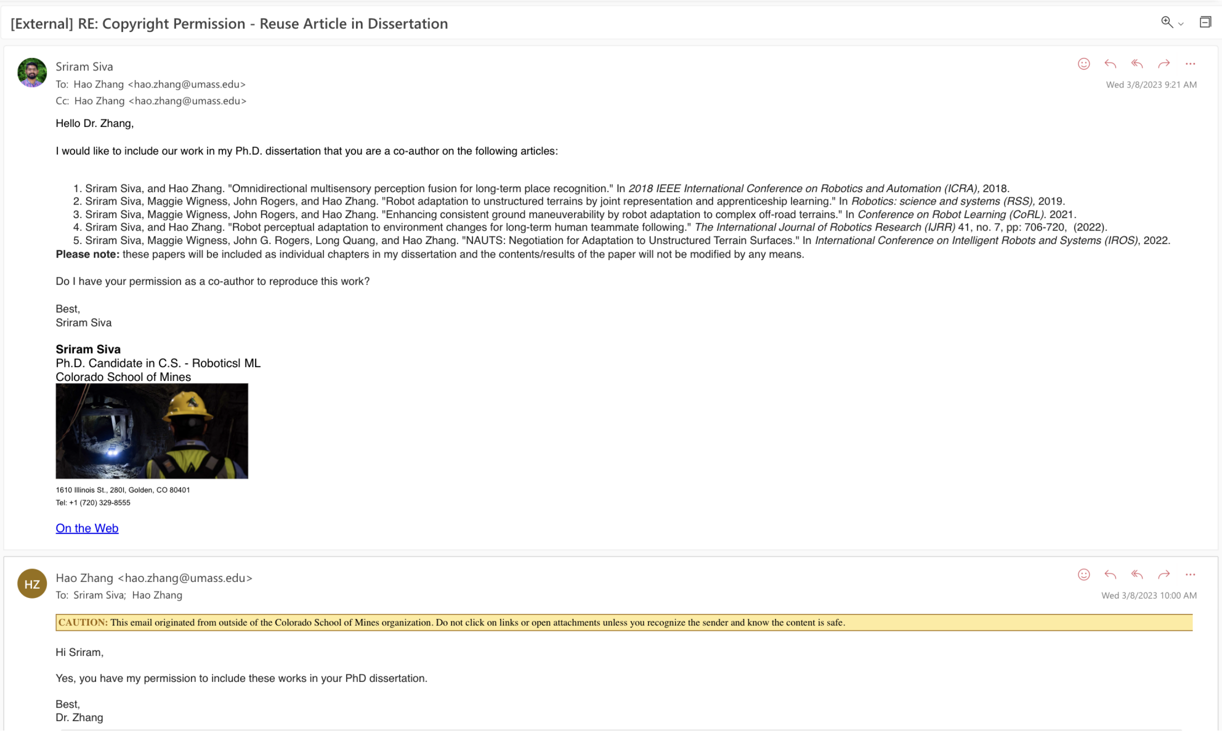
- IEEE reqires that "authors may reproduce or authorize others to reproduce the work" but requires a note of IEEE copyright.
- RSS has a policy on accepted works that "the copyright remains with the authors".
- Sage states that authors can use the published paper "in your dissertation or thesis, including where the dissertation or thesis will be posted in any electronic Institutional Repository or database".
- CoRL requires "Author retains the right to republish the contribution in any collection consisting solely of Author's own works without charge, subject to ensuring that the publication of the Publisher is properly credited and that the relevant copyright notice is repeated verbatim".

The copyright notices of IEEE, RSS, Sage and CoRL have been included in the footnotes of the relevant chapters.

A.2 Permission from Co-Authors

Reproduced works in this dissertation were submitted with multiple co-authors. Their written permission to include these chapters is included as screenshots of emails.

- Permission from Dr. Hao Zhang (Figure A.1) was granted to reproduce Chapters 2 to 8.
- Permission from Dr. Maggie Wigness (Figure A.2) was granted to reproduce Chapters 4, 7, and 5.
- Permission from Dr. John Rogers (Figure A.3) was granted to reproduce Chapters 4, 7, and 5.
- Permission from Long Quang (Figure A.4) was granted to reproduce Chapter 5.



[External] RE: Copyright Permission - Reuse Article in Dissertation

Sriram Siva
To: Hao Zhang <hao.zhang@umass.edu>
Cc: Hao Zhang <hao.zhang@umass.edu>
Wed 3/8/2023 9:21 AM

Hello Dr. Zhang,

I would like to include our work in my Ph.D. dissertation that you are a co-author on the following articles:

1. Sriram Siva, and Hao Zhang. "Omnidirectional multisensory perception fusion for long-term place recognition." In *2018 IEEE International Conference on Robotics and Automation (ICRA),* 2018.
2. Sriram Siva, Maggie Wigness, John Rogers, and Hao Zhang. "Robot adaptation to unstructured terrains by joint representation and apprenticeship learning." In *Robotics: science and systems (RSS),* 2019.
3. Sriram Siva, Maggie Wigness, John Rogers, and Hao Zhang. "Enhancing consistent ground maneuverability by robot adaptation to complex off-road terrains." In *Conference on Robot Learning (CoRL).* 2021.
4. Sriram Siva, and Hao Zhang. "Robot perceptual adaptation to environment changes for long-term human teammate following." *The International Journal of Robotics Research (IJRR)* 41, no. 7, pp: 706-720, (2022).
5. Sriram Siva, Maggie Wigness, John G. Rogers, Long Quang, and Hao Zhang. "NAUTS: Negotiation for Adaptation to Unstructured Terrain Surfaces." In *International Conference on Intelligent Robots and Systems (IROS)*, 2022.

Please note: these papers will be included as individual chapters in my dissertation and the contents/results of the paper will not be modified by any means.

Do I have your permission as a co-author to reproduce this work?

Best,
Sriram Siva

Sriram Siva
Ph.D. Candidate in C.S. - Roboticsl ML
Colorado School of Mines

1610 Illinois St., 280I, Golden, CO 80401
Tel: +1 (720) 329-8555

On the Web

Hao Zhang <hao.zhang@umass.edu>
To: Sriram Siva; Hao Zhang
Wed 3/8/2023 10:00 AM

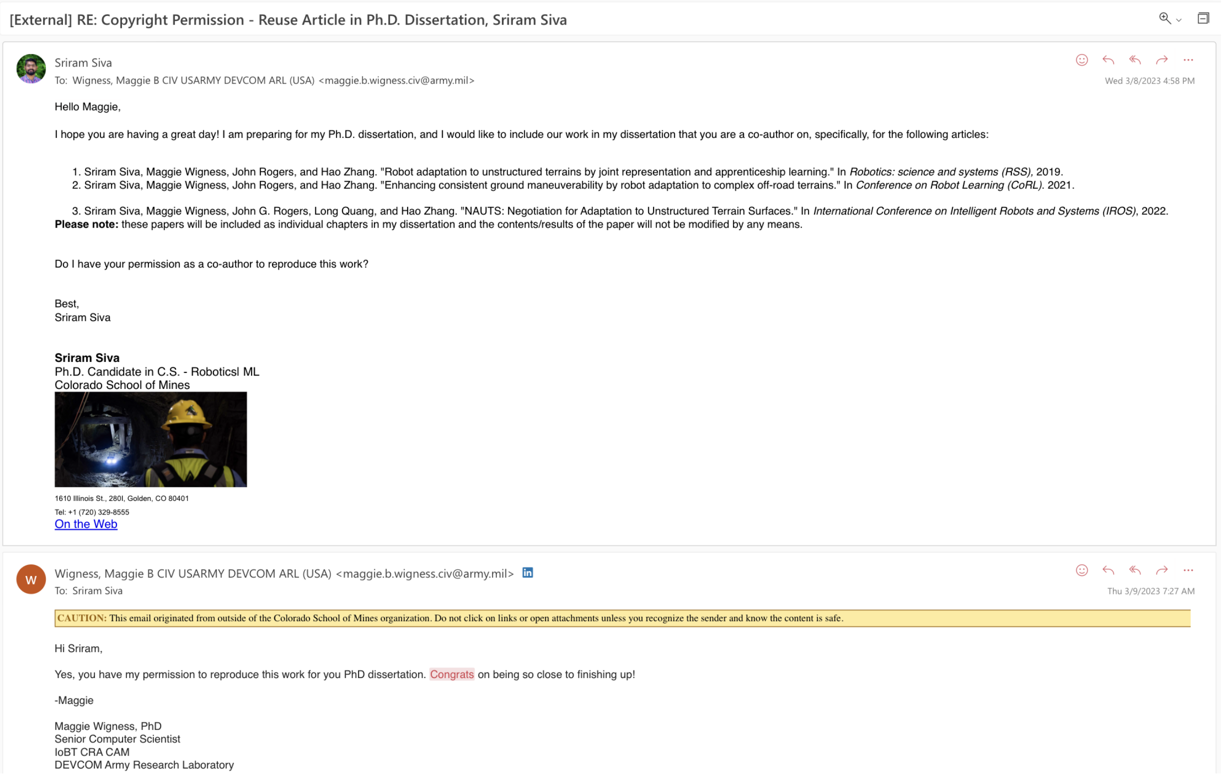
CAUTION: This email originated from outside of the Colorado School of Mines organization. Do not click on links or open attachments unless you recognize the sender and know the content is safe.

Hi Sriram,

Yes, you have my permission to include these works in your PhD dissertation.

Best,
Dr. Zhang

Figure A.1 Copyright permission from Dr. Hao Zhang.



[External] RE: Copyright Permission - Reuse Article in Ph.D. Dissertation, Sriram Siva

Sriram Siva
To: Wigness, Maggie B CIV USARMY DEVCOM ARL (USA) <maggie.b.wigness.civ@army.mil>
Wed 3/8/2023 4:58 PM

Hello Maggie,

I hope you are having a great day! I am preparing for my Ph.D. dissertation, and I would like to include our work in my dissertation that you are a co-author on, specifically, for the following articles:

1. Sriram Siva, Maggie Wigness, John Rogers, and Hao Zhang. "Robot adaptation to unstructured terrains by joint representation and apprenticeship learning." In *Robotics: science and systems (RSS)*, 2019.
2. Sriram Siva, Maggie Wigness, John Rogers, and Hao Zhang. "Enhancing consistent ground maneuverability by robot adaptation to complex off-road terrains." In *Conference on Robot Learning (CoRL)*. 2021.
3. Sriram Siva, Maggie Wigness, John G. Rogers, Long Quang, and Hao Zhang. "NAUTS: Negotiation for Adaptation to Unstructured Terrain Surfaces." In *International Conference on Intelligent Robots and Systems (IROS)*, 2022.

Please note: these papers will be included as individual chapters in my dissertation and the contents/results of the paper will not be modified by any means.

Do I have your permission as a co-author to reproduce this work?

Best,
Sriram Siva

Sriram Siva
Ph.D. Candidate in C.S. - Roboticsl ML
Colorado School of Mines

1610 Illinois St., 280I, Golden, CO 80401
Tel: +1 (720) 329-8555
On the Web

Wigness, Maggie B CIV USARMY DEVCOM ARL (USA) <maggie.b.wigness.civ@army.mil>
To: Sriram Siva
Thu 3/9/2023 7:27 AM

CAUTION: This email originated from outside of the Colorado School of Mines organization. Do not click on links or open attachments unless you recognize the sender and know the content is safe.

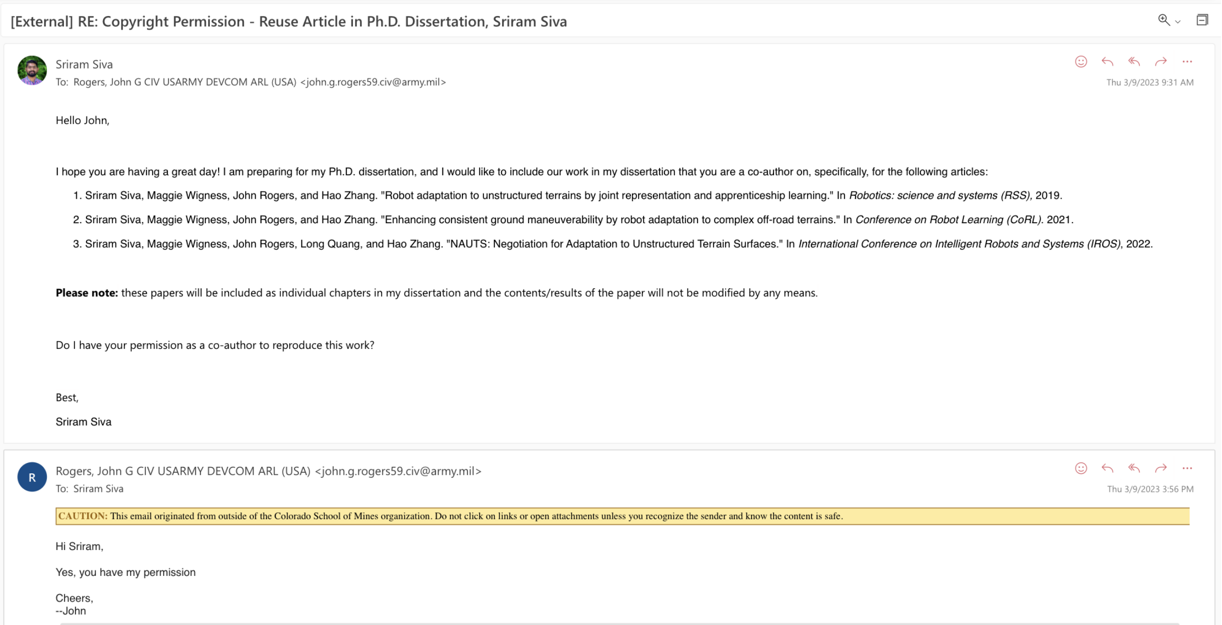
Hi Sriram,

Yes, you have my permission to reproduce this work for you PhD dissertation. Congrats on being so close to finishing up!

-Maggie

Maggie Wigness, PhD
Senior Computer Scientist
IoBT CRA CAM
DEVCOM Army Research Laboratory

Figure A.2 Copyright permission from Dr. Maggie Wigness.



[External] RE: Copyright Permission - Reuse Article in Ph.D. Dissertation, Sriram Siva

Sriram Siva
To: Rogers, John G CIV USARMY DEVCOM ARL (USA) <john.g.rogers59.civ@army.mil>
Thu 3/9/2023 9:31 AM

Hello John,

I hope you are having a great day! I am preparing for my Ph.D. dissertation, and I would like to include our work in my dissertation that you are a co-author on, specifically, for the following articles:

1. Sriram Siva, Maggie Wigness, John Rogers, and Hao Zhang. "Robot adaptation to unstructured terrains by joint representation and apprenticeship learning." In *Robotics: science and systems (RSS)*, 2019.
2. Sriram Siva, Maggie Wigness, John Rogers, and Hao Zhang. "Enhancing consistent ground maneuverability by robot adaptation to complex off-road terrains." In *Conference on Robot Learning (CoRL)*. 2021.
3. Sriram Siva, Maggie Wigness, John Rogers, Long Quang, and Hao Zhang. "NAUTS: Negotiation for Adaptation to Unstructured Terrain Surfaces." In *International Conference on Intelligent Robots and Systems (IROS)*, 2022.

Please note: these papers will be included as individual chapters in my dissertation and the contents/results of the paper will not be modified by any means.

Do I have your permission as a co-author to reproduce this work?

Best,

Sriram Siva

Rogers, John G CIV USARMY DEVCOM ARL (USA) <john.g.rogers59.civ@army.mil>
To: Sriram Siva
Thu 3/9/2023 3:56 PM

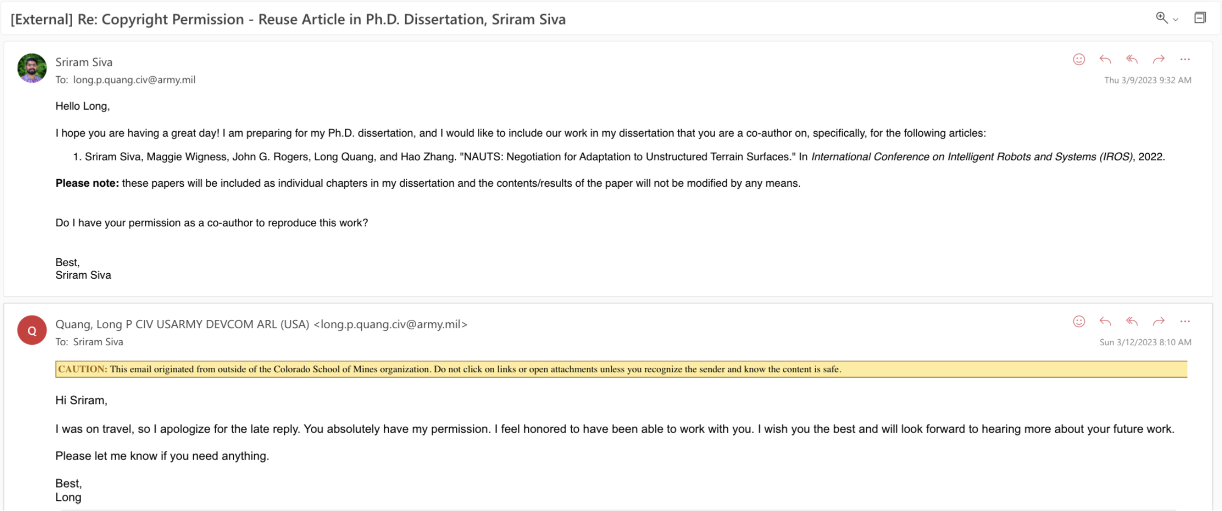
CAUTION: This email originated from outside of the Colorado School of Mines organization. Do not click on links or open attachments unless you recognize the sender and know the content is safe.

Hi Sriram,

Yes, you have my permission

Cheers,
--John

Figure A.3 Copyright permission from Dr. John Rogers.



[External] Re: Copyright Permission - Reuse Article in Ph.D. Dissertation, Sriram Siva

Sriram Siva
To: long.p.quang.civ@army.mil
Thu 3/9/2023 9:32 AM

Hello Long,

I hope you are having a great day! I am preparing for my Ph.D. dissertation, and I would like to include our work in my dissertation that you are a co-author on, specifically, for the following articles:

1. Sriram Siva, Maggie Wigness, John G. Rogers, Long Quang, and Hao Zhang. "NAUTS: Negotiation for Adaptation to Unstructured Terrain Surfaces." In *International Conference on Intelligent Robots and Systems (IROS)*, 2022.

Please note: these papers will be included as individual chapters in my dissertation and the contents/results of the paper will not be modified by any means.

Do I have your permission as a co-author to reproduce this work?

Best,
Sriram Siva

Quang, Long P CIV USARMY DEVCOM ARL (USA) <long.p.quang.civ@army.mil>
To: Sriram Siva
Sun 3/12/2023 8:10 AM

CAUTION: This email originated from outside of the Colorado School of Mines organization. Do not click on links or open attachments unless you recognize the sender and know the content is safe.

Hi Sriram,

I was on travel, so I apologize for the late reply. You absolutely have my permission. I feel honored to have been able to work with you. I wish you the best and will look forward to hearing more about your future work.

Please let me know if you need anything.

Best,
Long

Figure A.4 Copyright permission from Long Quang.