Embracing Uncertainty: Resilient Robotic Autonomy for Rough Terrain Exploration

February 2025

Masafumi Endo

A Dissertation for the Degree of Ph.D. in Engineering

Embracing Uncertainty: Resilient Robotic Autonomy for Rough Terrain Exploration

February 2025

Graduate School of Science and Technology
Keio University

Masafumi Endo

Only those who will risk going too far can possibly find out how far one can go.

— Thomas Stearns Eliot

Acknowledgments

First and foremost, I would like to express my deepest gratitude to Professor Genya Ishigami, my principal advisor. His thoughtful guidance and constant support throughout my doctoral studies have shaped not only my research direction but also my fundamental understanding of how to identify and pursue meaningful research questions. His careful mentorship, clear vision for research, and encouragement to explore new opportunities have given me both the foundation and confidence to grow as a researcher.

I am deeply grateful to my dissertation committee members: Professor Yoshimitsu Aoki, Professor Komei Sugiura, and Professor Masaki Takahashi. Their expertise in different fields helped me understand the fundamental challenges in robotics and beyond. Their thoughtful feedback has greatly improved this dissertation while providing valuable insights that will guide my future research endeavors.

I would like to especially thank Dr. Masahiro Ono, who not only served on my dissertation committee but also welcomed me as an intern at NASA JPL. Ten years ago, through his book, he first showed me the exciting world of space exploration, and his passion deeply inspired me. During my internship, his dedicated mentorship and our daily discussions led to many meaningful research outcomes. I am grateful to Dr. Ashkan Jasour and Rohan Thakker for their valuable insights into risk-aware autonomy theory throughout my time at JPL. Working with the EELS team, especially Dr. Michael Paton, Dr. Guglielmo Daddi, Dr. Michael Swan, and many others, has been one of the highlights of my doctoral journey. My thanks also go to Shigeru Suzuki, who made my time in Pasadena feel like home.

I would like to thank Dr. Ryo Yonetani and Dr. Tatsunori Taniai for welcoming me, a researcher from field robotics, during my time at OMRON SINIC X. They shared their cutting-edge insights on bridging machine learning and path planning. I am particularly grateful to Dr. Yonetani for showing me his vision for advancing robotics through AI and for his leadership in realizing it. Dr. Taniai's ability to deeply engage with every challenge, no matter how trivial, and find in-depth solutions through his thorough understanding, has deeply influenced my approach to research.

Their guidance has set standards I will always strive towards.

I want to thank Dr. David D. Fan and Dr. Ali Agha for guiding me during my research internship at Field AI. They taught me how to bridge the gap between theoretical research and real-world robotics. Working hands-on with quadruped robots and developing autonomous systems showed me the engineering challenges we face and the real impact robots can have in the world. Their guidance helped me understand what it takes to make robotics work in practice. Special thanks to Dr. Oriana Peltzer for leading field operations in Japan and always being there with her warm support.

I want to thank Dr. Kohei Honda from Nagoya University for his collaboration on our motion planning project for off-road vehicles. As a peer researcher at the forefront of the field, his willingness to collaborate has meant a lot to me. I admire his approach to finding theories from practical applications and extending them to new domains like field robotics.

The Space Robotics Group (SRG) at Keio University has been my academic home these past years. I especially want to thank my fellow Ph.D. students, Takuya Omura, Ryuki Shimada, Dr. Reiya Takemura, Dr. Go Sakayori, Taisei Nishishita, and Sora Ishikawa, who shared countless discussions and challenges throughout this journey. I am particularly grateful to Takuya Omura for sharing these three years with me and teaching me the value of supporting others in need. A special thanks to Takahiro Fuke: while this was my first experience providing technical guidance, his achievements far exceeded my expectations and showed me the joy of mentorship. I look forward to seeing the future achievements of Aman Arora, Risa Ito, and Nobuaki Aoki in their professional research careers at their new institutions. I would also like to thank all the other members, especially Yuki Goto, who made my daily life at SRG truly enjoyable and memorable.

My time at the Space Robotics Laboratory (SRL) at Tohoku University, where I earned my Master's degree, laid the foundation for my doctoral studies. I am grateful to my former supervisors, Professor Kazuya Yoshida, Professor Kenji Nagaoka, and Professor Toshinori Kuwahara, who gave me the freedom to pursue my research interests even when results were hard to come by. This experience helped me develop the independence I needed for my doctoral studies. I would also like to extend my gratitude to all members of SRL. Particularly, Professor Kentaro Uno, Dr. Yuki Shirai, and Dr. Shoya Higa have always inspired me as peers in robotics research and set standards I aspire to reach. I look forward to our future encounters and collaborations.

Throughout my academic journey, I have been fortunate to meet many inspiring researchers. I am grateful to my early career mentors from different fields: Professor Tsuyoshi Kawanami at Kobe University, who guided my undergraduate studies, and Professor Burak Sencer at Oregon State University, who supervised my master's research. I have also been inspired by world-leading researchers, particularly Takahiro Miki from ETH Zurich and Dr. Haruki Nishimura from Toyota Research Institute, who shared their valuable insights during seminars at Keio University. These encounters at conferences and seminars have broadened my perspective and shown me the diverse paths in research.

I am deeply grateful for the financial support that made this research possible. The Japan Society for the Promotion of Science (JSPS) provided me with the Research Fellowship for Young Scientists (DC1), which allowed me to focus fully on my research, and the Overseas Challenge Program for Young Researchers, which enabled my research stay at NASA JPL. I would also like to thank Keio University for its various scholarships and the JEES/Mitsubishi Corporation Scholarship program for its long-term support. These generous funding opportunities allowed me to dedicate myself to research without financial concerns.

I would like to express my heartfelt gratitude to my parents, Keisuke Endo and Kayoko Endo. Your unwavering presence and support throughout this journey have given me the strength to persevere and achieve my goals.

Finally, to my partner - thank you for always being by my side. Nothing happened without you.

February 2025
Masafumi Endo

Thesis Abstract

How can mobile robots explore the unknown? This dissertation addresses the fundamental question of achieving resilient robotic autonomy in challenging open-world scenarios. During rough terrain exploration, geometric and non-geometric hazards present substantial risks of robot damage and mission failure. Robust perception of the environment is critical for identifying potential hazards to support informed decisions for persistent navigation. However, ever-present *uncertainty* in perception leads to flawed risk assessment and unsafe decisions that threaten robots operating in unstructured environments. While robot safety can be ensured through overly conservative operations or careful human supervision, such a "chained" approach limits autonomous operations and prolongs mission timelines. Addressing uncertainty is thus key to realizing the full potential of robotic autonomy for faster and more extensive exploration.

This dissertation establishes an approach toward embracing uncertainty through perception and planning algorithms for achieving resilient off-road autonomy in rough terrain exploration. Uncertainty on rough terrain stems from complex robot-terrain interactions captured through proprioceptive and exteroceptive sensing. The proposed methods introduce mathematical frameworks to *quantify* two categories of uncertainty: scene understanding for planning and state transitions for execution. Building on this quantification, the presented path and motion planning algorithms leverage uncertainty through *exploitation* to avoid mission-critical risks and *adaptation* to reduce perceptual uncertainties, enabling safe and time-efficient navigation.

The first part of this dissertation addresses uncertainty in scene understanding, arising from non-geometric hazards on deformable surfaces that induce wheel slip and can ultimately immobilize a robot. Such hazards introduce predictive uncertainty into machine learning (ML)-based traversability prediction, resulting in erroneous decision-making. The study begins with *deterministic* scene understanding, where combining multiple ML models—from terrain type classification to slip trend regression—inherently propagates erroneous predictions. The proposed terrain-dependent slip prediction method integrates distinct ML models while accounting for uncertainty when

converting classification outputs into regression inputs. Experimental validation with real-world data demonstrates robust traversability predictions despite uncertainty in exteroceptive perception.

This study then focuses on *probabilistic* scene understanding to *quantify* predictive uncertainty as a probability distribution. By employing Gaussian processes, the first part proposes *active traversability learning*, a framework in which risk-taking planning guides robots toward hazardous yet informative slip experiences. The developed method integrates informative path planning with chance constraints for risk evaluation over slip distributions, enabling aggressive maneuvers for adaptation while preventing rover entrapment. The subsequent work proposes *probabilistic fusion* of distinct ML models for terrain classification and slip regression into a unified probability distribution. This multimodal slip distribution preserves predictive uncertainties and supports risk-aware planning through statistical risk metrics. These uncertainty exploitation and adaptation are finally integrated into a unified learning and planning framework for resilient rover navigation. The core concept, *deep probabilistic traversability*, is an end-to-end probabilistic ML model that learns slip distributions directly from in-situ traverse data, thus enabling simultaneous uncertainty handling under self-supervision. Extensive evaluations on rigorous synthetic datasets demonstrate that embracing predictive uncertainty contributes to safer and more efficient rover navigation.

The last component of this dissertation addresses uncertainty in state transitions emerging during trajectory execution. Due to the inability to perform accurate state estimation and locomotion simultaneously, the proposed integrated task and motion planning (TAMP) framework jointly optimizes state estimation schedules along with motion planning. A novel *continuous-time* planning approach is developed to avoid time discretization, significantly reducing complexity and enabling rapid computation. By adopting a robust approach to modeling uncertainty growth, TAMP is formulated as mixed-integer linear programming that guarantees robot safety under worst-case conditions while scheduling intermittent scanning. Validation through simulation and hardware experiments demonstrates robust and efficient navigation even with limited perceptual feedback.

The perception and planning algorithms developed in this dissertation embrace uncertainty by transforming unknowns into informed decisions that contribute to safety and mission success in rough terrain exploration. The philosophy of *embracing uncertainty* will unlock reliable robotic autonomy in challenging open-world scenarios, where robots venture boldly into the unknown.

Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Background and Motivation

"Exploration is really the essence of the human spirit," said former National Aeronautics and Space Administration (NASA) astronaut Frank Borman. This fundamental drive to explore the unknown has been a defining trait of humanity and remains crucial to expanding the noosphere. However, the pursuit of exploration often leads us to environments where direct human presence is either impractical or poses substantial safety risks. Planetary surface exploration represents one of the most extreme cases of such challenges, which brings opportunities for both civilizational advancement and scientific discoveries while presenting significant constraints for sustaining human life. Mobile robots have emerged as key substitutes for human explorers in such harsh terrains, due to their resilience under conditions hostile to life, including extreme temperatures, vacuum, and radiation. In addition to safety benefits, these robots can reduce human exposure to labor-intensive activities under adverse conditions, such as long-term inspection and monitoring of infrastructure and natural environments. In this way, robotic technologies help alleviate human burdens associated with field operations, contributing to more sustainable and far-reaching exploration efforts.

In the early stages of robotics, mobile robots were operated through human operation, serving as prosthetic extensions of the human body. However, intensive human guidance to robots requires significant operational resources and faces inherent limitations in real-world deployments. Such constant reliance on human operators proves impractical or impossible in many field environments, where natural terrain lacks network infrastructure for continuous communication and inherent latency exists in interplanetary communications. These operational challenges, from communication latency to high operator workload, often compromise mission efficiency. Moreover, a nonnegligible gap exists between operators' understanding and robots' actual situation in rough terrain. This situational awareness gap leads to overlooking potential hazards such as natural rocks or terrain roughness, which may damage the robots and result in mission failures. To overcome these issues, robots must operate with a level of autonomy that enables appropriate situational awareness and informed, in-situ decision-making. Such autonomy supports safer and more efficient mission operations, particularly in the challenging environments at the frontier of exploration.

1.2 Problem Statement

Robotic autonomy arises from the continuous interaction between a robot and its environment. Here, *perception* from onboard sensors or external resources identifies both the robot's internal states and external conditions, while *planning* uses this information to decide actions that guide the robot toward its destination. Robots rely on two types of perception: proprioceptive sensors for measuring their own states, such as joint encoders and inertial measurement units (IMUs), and exteroceptive sensors for recognizing the surrounding environment, such as stereo cameras and light detection and ranging (LiDAR). However, ever-present and ineradicable *uncertainty* exists in robotic perception, which prevents robots from obtaining complete and accurate information about their states and surroundings. This uncertainty threatens robot safety as it causes a flawed evaluation of potential hazards and leads to decisions that do not cope with the actual situation. Particularly on unexplored rough terrain, considerable uncertainty is present in complex robot-terrain interactions captured through proprioceptive and exteroceptive perceptions. Understanding the nature of the uncertainties in rough terrain exploration requires distinguishing between two categories of uncertainty [3]:

- **Aleatoric uncertainty**: This category arises from inherent randomness or variability in the environment and sensing processes. Examples include stochastic sensor noise and unpredictable terrain behavior, such as irregular soil deformation. Aleatoric uncertainty cannot be reduced by gathering additional data; it reflects the intrinsic variability of the phenomena the robot is trying to model.

- **Epistemic uncertainty**: This category stems from a lack of knowledge or incomplete understanding of the system and its environment. Examples include inaccurate terrain representations from partial observations or simplified models of robot-terrain interactions. Unlike aleatoric uncertainty, epistemic uncertainty can be reduced through improved modeling techniques, more extensive data collection, or refined estimation algorithms.

Given that these uncertainties are inherent in robotic systems, this dissertation argues that achieving reliable autonomy in rough terrain requires not only recognizing their presence but also incorporating them into perception and planning processes, rather than seeking unrealistic certainty. By representing uncertainty and integrating it into decision-making, robots can transform unknowns into informed actions that contribute to both robot safety and mission efficiency, even in challenging open-world scenarios.

1.3 Objective

The objective of this dissertation is to address the uncertainty that arises from complex robot-terrain interactions by establishing autonomy methodologies, as discussed in the previous section. Rather than attempting to eliminate uncertainty, we seek to *embrace* uncertainty—understanding the underlying unknowns within perceptual data and leveraging them to make more informed decision-making. Our approach centers on two key autonomy components: perception, which enables the quantification of uncertainty, and planning, which exploits this quantification to guide long-duration operations over challenging terrain. Achieving reliable autonomy in such environments requires *resilience*, encompassing both robustness to cope with inherent, aleatoric randomness and adaptability to refine understanding and address epistemic knowledge gaps. Accordingly, this dissertation presents perception and planning algorithms that not only withstand potential hazards but also adapt to novel conditions, thus enabling safer and more efficient exploration of rough terrain.

1.4 Scope

Rough terrain exploration encompasses various applications for modern society, such as agriculture [4], environmental monitoring [5], facility inspection, subterranean exploration [6, 7], search-and-rescue [8], and planetary exploration missions [9]. The physical characteristics of these terrains—ranging from complex surface geometries to extreme environmental conditions—combined with demanding mission requirements, such as energy efficiency for long-term operation and robust mobility in challenging environments, necessitate careful consideration in the selection of mobility mechanisms. To address these challenges, various robotic mechanisms have been developed for rough terrain exploration, each tailored for environmental conditions and mission objectives. These include quadruped robots designed for geometrically complex surfaces [10], climbing robots for vertically challenging terrains [11], hopping robots for low gravity environments [12], snake-type robots for confined spaces [13], and wheeled robots for energy-efficient operation in rough terrain scenarios [14].

This dissertation specifically focuses on achieving resilient robotic autonomy in *planetary surface exploration*, where significant communication delays between Earth and space demand robust and self-reliant decision-making. Mobile robots, or rovers, must navigate rough terrain that poses both geometric and non-geometric hazards, each of which carries a risk of damage and mission failure. Previous planetary exploration missions (*e.g.*, the Mars rovers [15]) employed onboard perception systems to detect and avoid geometric obstacles to enable basic autonomy. Yet these missions still required frequent human intervention to evaluate the ability to traverse, *i.e.*, *traversability* in unexplored environments, resulting in slow and constrained operations.

At the heart of these limitations lies the inherent *uncertainty in robotic perception.* Sensor noise, incomplete environmental models, and complex interactions between robot and terrain prevent robots from attaining perfect situational awareness. This uncertainty comes in the form of two major challenges:

- **Uncertainty in scene understanding for planning**: Non-geometric hazards associated with deformable surfaces introduce significant ambiguity in predicting robot-terrain interactions. Machine learning (ML) is a powerful tool for traversability prediction by discovering latent terrain properties from training samples, but no ML model guarantees perfect predictions due to inherent *predictive uncertainty.* This uncertainty hinders confident risk assessment for decision-making, requiring either overly conservative strategies or human supervision.

- **Uncertainty in state transitions for execution**: Even after planning sequential decisions, accurate execution of the planned trajectory is difficult due to uncertainties that propagate over time in state estimation and vehicle dynamics. Limited perceptual feedback or intermittent sensing can cause the robot's actual trajectory to deviate from the intended path, increasing the risk of approaching hazards such as entrapment and collisions.

In this dissertation, we address these two key challenges—uncertainty in scene understanding and state transitions—as fundamental issues arising from imperfect robotic perception. We develop a set of perception and planning algorithms that *embrace* these uncertainties, rather than merely tolerating them. Our goal is to establish reliable rover navigation from start to destination, a core requirement for autonomous planetary surface exploration. By quantifying uncertainty, we enable two complementary strategies in planning: exploiting it to avoid potential hazards and adapting our models and plans to reduce it over time. These strategies work together to improve rover autonomy and thus allow more independent and reliable operations in challenging planetary environments.

1.5 Structure

The structure of this dissertation is summarized in Fig. 1.1. Chapter 1 highlights the importance of addressing uncertainty in robotic perception, divided into scene understanding for planning and state transitions for execution, to achieve reliable robotic autonomy in the field. Building on the principle of uncertainty quantification, this dissertation aims to develop resilient rover navigation for planetary exploration, ensuring that the perception and planning algorithms provide robustness by maintaining safe operation despite uncertain hazards and adaptability by refining its models or strategies to handle novel scenarios as they arise.

We begin by addressing challenges where uncertainty arises in scene understanding for planning, particularly in predicting the traversability of wheeled rovers using ML models. Chapter 2 focuses on quantifying uncertainty derived from perception based on deterministic ML models, which integrate their outputs to predict wheel slip across heterogeneous terrain. From Chapter 3 onwards, we incorporate probability theory to represent predictive uncertainty as a probability distribution, to achieve well-calibrated uncertainty quantification and to enable its exploitation in planning, both of which persist through subsequent chapters. Chapter 3 investigates the adaptability of probabilistic ML models by employing a robotic information-gathering algorithm to guide rovers toward informative traversal experiences. In contrast, Chapter 4 addresses robustness against the inevitable prediction errors of ML models for planetary rover path planning by introducing a novel traversability representation and evaluating costs

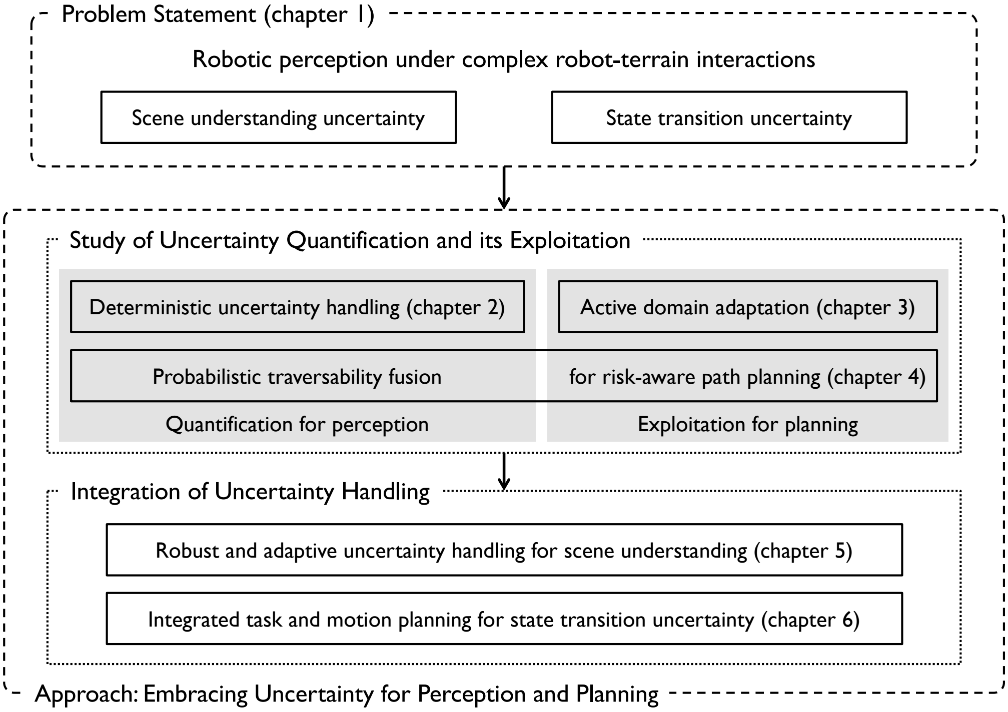


Fig. 1.1: Overview of the structure of this dissertation, beginning with the problem statement and leading to the principal approach, which embraces uncertainty in robotic perception by quantifying it and then exploiting it for planning.

using statistical risk metrics. Finally, Chapter 5 integrates these two components for resilience into a unified learning and planning framework owing to an end-to-end probabilistic ML model that supports robust and adaptive navigation through self-supervised traversability learning.

Next, we turn our attention to uncertainty arising in state transitions during execution, focusing on snake robots that possess limited capabilities for simultaneous motion and accurate localization using exteroceptive sensors. Chapter 6 addresses resilient navigation in the presence of localization failures by proposing blind motion with intermittently scheduled scans (BLISS). BLISS achieves both robustness against potential collisions and adaptability to uncertain states by balancing conservative blind locomotion and periodic scanning for accurate localization. Although this approach leads to a computationally intractable TAMP problem under uncertainty, we overcome this challenge by proposing a continuous-time planning formulation that significantly reduces the complexity of the problem.

Finally, Chapter 7 concludes by summarizing the significance of the philosophy of this dissertation—*embracing uncertainty*—for resilient robotic autonomy in rough terrain exploration. We review the proposed key methods and discuss future research directions, including improvements to perception and planning algorithms to realize trustworthy robotic autonomy in the field.

1.6 Contributions

This dissertation advances conventional autonomous navigation by incorporating uncertainty awareness into robotic perception and planning, which enables rovers to navigate more safely and efficiently through unfamiliar environments. Our framework for resilient off-road autonomy transforms uncertainty from a limiting factor into a valuable resource for robust and adaptive robotic exploration. The specific contributions of this dissertation are summarized as follows. The references accompanying each item represent the author's work published, under review, or in preparation in the refereed conference proceedings and journals.

- **Deterministic uncertainty handling for traversability prediction [4]**: We introduce a deterministic traversability prediction method that integrates multiple ML models that incorporate both visual and geometric information from natural terrain. Visual features are employed for terrain type classification, while geometric features model slope-dependent slip behavior for each terrain category. The contribution lies in integrating multiple ML models while accounting for uncertainty arising from discrepancies between actual and estimated terrain geometry from stereo vision. This integration enables conservative yet reliable slip risk predictions tailored to each terrain class even in unstructured environments. The effectiveness of the proposed traversability prediction method is validated using real-world datasets obtained from a rover testbed.
- **Active domain adaptation via robotic information gathering [2, 10]**: We present a method for learning traversability in novel environments by leveraging a Gaussian process (GP) to represent uncertainty as a probability distribution. The contribution is a novel decision-making strategy that actively improves traversability models through deliberate, risk-taking maneuvers. We implement this concept as risk-aware informative path planning that incorporates probabilistic constraints to mitigate the risk of rover entrapment. A sampling-based planning algorithm is developed to solve the problem of generating trajectories to informative locations. Through extensive simulations in rough terrain environ-

ments, we demonstrate that our active domain adaptation approach effectively reduces predictive uncertainty in ML-based traversability prediction.

- **Probabilistic traversability fusion for risk-aware path planning** [9]: We propose a risk-aware path planning method to be robust against prediction errors in ML-based traversability models operating in heterogeneous terrain. The contribution is a probabilistic traversability representation that fuses multiple ML models into a single multimodal probability distribution. This method captures comprehensive uncertainties from different models and supports consistent exploitation of predictive uncertainty for conservative path planning using statistical risk metrics. Through extensive simulations in rough terrain environments with varying degrees of prediction complexity, we demonstrate that the proposed method generates more feasible paths compared to existing approaches, ultimately improving the reliability and safety of navigation.

- **Robust and adaptive uncertainty handling for scene understanding** [1]: We integrate three principal approaches to predictive uncertainty in scene understanding addressed so far—uncertainty quantification, uncertainty exploitation, and model adaptation—into a unified learning and planning framework. The contribution is developing an end-to-end probabilistic ML model using deep neural networks (DNNs) for traversability prediction. Leveraging this model, the proposed framework enables direct prediction of slip distributions from visual and geometric information for uncertainty quantification and exploitation in path planning, while allowing adaptation of pre-trained models based on in-situ traverse experience. Through extensive simulations across rough terrain environments encompassing various out-of-domain scenarios, we demonstrate that the proposed method achieves robust and adaptive navigation, leading to safer and more efficient autonomous operation compared to existing approaches.

- **Integrated task and motion planning for state transition uncertainty** [5, 6]: We present a resilient navigation method to prevent obstacle collisions due to localization failures in rovers exploring distant planets, which have limited capabilities for simultaneous motion and accurate localization. We establish a new task and motion planning (TAMP) approach that jointly plans blind motion with intermittently scheduled scans to balance the trade-off between ensuring collision safety and maintaining efficiency, despite the time-consuming exteroceptive localization. Our contribution is a novel continuous-time planning formulation that avoids time discretization in computationally challenging integrated TAMP under uncertainty. This formulation accelerates TAMP computation by significantly reducing the number of optimization variables in NP-hard mixed-integer linear programming. Through both simulations and real-world experiments using a snake-like robot named Exobiology Extant Life Surveyor (EELS), we demonstrate that the proposed method achieves more rapid TAMP than existing approaches, enabling resilient navigation in obstacle-rich, unstructured environments.

Chapter 2

Deterministic Uncertainty Handling for Traversability Prediction

2.1 Introduction

Planetary exploration rovers have been used for conducting detailed investigations of extraterrestrial surfaces such as those of Moon and Mars. During these missions, rovers have to be operated with weak communication signals and nonnegligible communication lag. To overcome these constraints, rovers have been required to automatically recognize their surrounding environment, detect obstacles, and travel through uncharted regions. For example, the Spirit and Opportunity rovers (National Aeronautics and Space Administration (NASA)) used conventional perception strategies that evaluated their surrounding environments by using stereo vision and detecting geometric obstacles [15]. Although this approach enabled successful long-term operation of the rovers, loose, granular materials on celestial surfaces can make the rover's wheels slip and, in the worst case, can cause the rover to get stuck without the possibility of recovery. In the Mars Science Laboratory (MSL) mission, Curiosity experienced excessive wheel slip in the Hidden Valley (Fig. 2.1) [16]. The terrain here comprised rippled sand that seemed safe and nonhazardous; nonetheless, Curiosity was forced to avoid these nongeometric obstacles.

To assess wheeled robots' mobility on rough terrain, the mechanical phenomena occurring in wheel-terrain interaction have been studied in the field of terramechanics [17]. Slip causes a lack of movement when wheeled robots traverse terrain. Although considering wheel slip is crucial for rovers, the complexity of modeling the highly nonlinear

Fig. 2.1: The Hidden Valley on Mars ©NASA/JPL-Caltech. Curiosity experienced excessive wheel slip here and failed to traverse this region although it is relatively flat. At the beginning of the mission, Curiosity was commanded to travel on the flat surface and avoid obstacles such as rocks; however, wheel slip occurred owing to loose sand with megaripples. Wheel traces on these megaripples indicated that Curiosity gave up traversing this surface and turned back to avoid getting stuck.

characteristics of rover-terrain interaction makes it difficult to evaluate and predict rovers' traversability to accomplish autonomous operations.

Visual information provides rovers with more clues to predict wheel slip from a distance. Geometric information of the terrain topography, such as steep slopes that cause wheel slip [18], can be obtained from 3D measurements through stereo vision. Further, visual characteristics such as color and texture provide semantic descriptions of terrain types to evaluate rovers mobility. Both of these clues are obtained from a distance; hence, exteroceptive sensing can potentially be used to predict wheel slip before entering the hazardous area by estimating slopes and assessing terrain types for rovers operating on rough terrain.

This study proposes a method to predict the wheel slip risk for a rover operating on rough terrain. This method involves two procedures: (1) slope estimation from 3D information and (2) terrain classification from images by using a machine-learning classifier. A regression curve optimized with previously experienced data is used as a slip versus slope relation to predict the wheel slip risk corresponding to the estimated slope angle and classified terrain. The terrain-dependent slip risk, classified as low, medium, or high, is predicted using the slip versus slope relation in consideration of the slope estimation accuracy. Thresholds are set for each terrain type because different terrain types have different impacts on the rover. Experimental results obtained using a rover testbed operated on several terrain types are presented to validate the proposed approach.

The rest of this chapter is organized as follows. Section 2.2 defines wheel slip and discusses related work. Section 2.3 presents details of the proposed slip risk prediction approach. Section 2.4 describes the experiment for validating the proposed approach, including detailed information of the testbed, data collection, and experimental conditions. Section 2.5 presents experimental results validating the performance of the proposed slip risk prediction approach. Finally, Section 2.6 concludes this chapter.

2.2 Related Work

2.2.1 Background on Slip

Wheel slip is considered one of the critical factors for rovers to traverse rough terrain. Wheel slip is quantitatively described by the slip ratio [19]. The slip ratio s is a proportion of the desired and actual traveling velocities and is expressed as follows:

$$s = \frac{r\omega - v}{r\omega} = 1 - \frac{v}{r\omega}(0 \leq s \leq 1), \tag{2.1}$$

where v is the actual traveling velocity, and r and ω are the radius and angular velocity of the wheel, respectively, as shown in Fig. 2.2. The slip ratio represents the degree of longitudinal slip. A positive slip ratio implies that a rover is traveling slower than commanded, and a slip ratio of 1 implies that the rover is completely stuck in rough terrain.

2.2.2 Traversability Assessment

Mobile robots must have vision so that they can assess their traversability. NASA's Mars Exploration Rover (MER) mission conducted geometric analyses using a stereo camera that detected natural obstacles, such as small rocks, to assess whether the rover's wheels could get over them [15]. However, this approach could not solve problems caused by rough terrain, such as wheel slip, sinkage, and complete immobilization of the rover, because geometric information does not contain semantic characterizations of the physical properties of the target terrain.

Learning-based terrain assessment approaches using visual information have been actively studied to enable mobile robots to detect such nongeometric hazards and avoid wheel slip on rough terrain. These approaches afford two main advantages: traversability can be predicted to detect potential risks before entering the region in front of the robots and modeling of complex wheel-terrain interactions can be avoided. Halatci et al. [20] presented a multisensor terrain classification method that applied Bayesian fusion of individual support vector machines (SVMs) using color, texture, and depth information acquired by stereo imagery. Brooks et al. [21] expanded this approach to self-supervised terrain classification by associating exteroceptive data with proprioceptive wheel vibration data. Otsu et al. [22] proposed co- and self-training approaches that classified the surrounding terrain of a rover with lesser image data. Several studies have also investigated traversability prediction as well as terrain classification. Berczi et al. [23] showed that Gaussian process (GP) classifiers enable learning and assessing the traversability of terrain with a high-dimensional representation. Schilling et al. [24] implemented a geometric and visual terrain classification method that predicts terrain traversability in a mixture of environments with less training data. Higa et al. [25] proposed a vision-based approach that remotely predicts the energy consumed by rovers when driving to search for and traverse desirable paths on rough terrain. Wheel slip is one of the indexes used to assess traversability. Angelova et al. [26] applied a mixture of expert models for combining the results of terrain classification and terrain-dependent slip prediction using locally weighted projection regression (LWPR) to predict wheel slip from visual information. Cunningham et al. [1] used GP regression for considering the uncertain wheel-terrain relation for each terrain, which was classified using convolutional neural networks (CNNs). Skonieczny et al. [27] evaluated the trafficability risk of planetary rovers using a slip-slope table specific to each terrain, which adapts to variable slip versus slope data, and by assigning thresholds between low, medium, and high slip risk. These thresholds were defined by considering the hazardous slip ratio to be 0.2; however, these fixed thresholds could cause incorrect wheel slip risk evaluation owing to the uncertainty of slope estimation. Although the slip risk monotonically increases with increasing slope, wheel slip trends differ with the terrain type

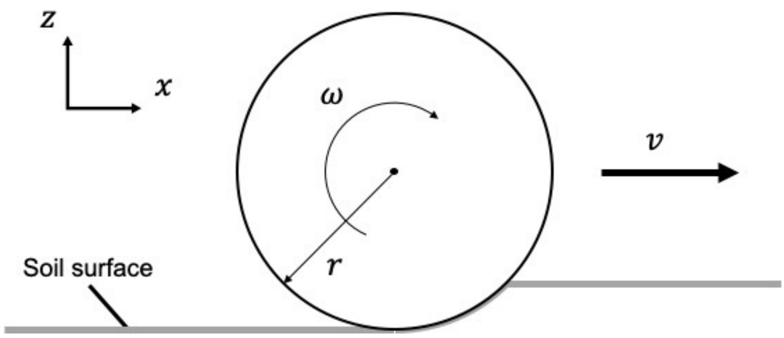


Fig. 2.2: Longitudinal wheel slip on loose soil.

being traversed. For instance, uneven gravel terrain causes a sudden increase in wheel slip, and the slip versus slope relation on terrain consisting of homogeneous loose sand is almost linear. Thus, terrain-dependent thresholds are a more reasonable criterion for evaluating the slip risk of wheeled robots on rough terrain.

This study predicts the slip risk of a rover traversing rough terrain. The main idea is to enable the rover to discover hazardous region from a distance. Two factors—terrain slope and terrain type—that affect the degree of wheel slip are investigated using visual information to predict wheel slip corresponding to the obtained slope angle and terrain classification. The proposed approach differs from the above-described methods in that it introduces terrain-dependent thresholds for the slip versus slope relation for the rover to predict the wheel slip risk. The mean absolute error (MAE) between the estimated slope angle and the ground truth value is used as a margin to bound the thresholds. By using this prediction method, (1) the characteristic wheel slip risk is predicted on the target region of the terrain and (2) the MAE enables avoiding underestimating the slip risk owing to the uncertain terrain surface.

2.3 Overview of Slip Risk Prediction

This section presents an overview of the proposed approach for learning the relation between visual information and wheel slip based on data obtained from both exteroceptive and proprioceptive sensors. Wheel slip results from wheel-terrain interactions, and the amount of wheel slip depends on geometric conditions such as the slope, state, and composition of the terrain. Therefore, the proposed slip risk prediction approach is based on slope estimation and terrain classification, as shown in Fig. 2.3, and the use of visual information from an RGB-D sensor. The slip risk is finally predicted by assessing the classified terrain and slip ratio corresponding to the estimated slope angle.

2.3.1 Slope Estimation

The slope estimation procedure acquires the slope angle of terrain in front of the rover by calculating the least squares plane using 3D point clouds acquired from an RGB-D sensor. To remove outliers in the point cloud data, the RANdomized SAmple Consensus (RANSAC) algorithm is applied as follows [28]:

1. Select n sample points at random from 3D point clouds $\mathcal{U}$.
2. Calculate the plane equation that minimizes the sum of squares of plane $\mathcal{Z}$ and n points.
3. Evaluate the coincidence between $\mathcal{U}$ and $\mathcal{Z}$ calculated in step 2.
4. Repeat from step 1 to obtain the least squares plane $\mathcal{Z}'$.
5. Calculate slope angle θ as the pitch angle for the rover coordinate $\sum_{\text{Rover}}\{X, Y, Z\}$ by using the following equation:

$$\theta = -\tan^{-1}\frac{\partial \mathcal{Y}}{\partial \mathcal{Z}'}. \tag{2.2}$$

This procedure also reflects the rover posture by acquiring data from inertial measurement units (IMUs) and potentiometers in the rover's suspension mechanism, as discussed in Section 4. The region of interest (ROI) for slope estimation is 0.8 m^2 in front of the rover, as shown in Fig. 2.4, considering the effective measurement range of the RGB-D sensor.

2.3.2 Terrain Classification

To assess the distinctive terrain type of a given region, terrain is classified by a machine-learning classifier using images as the input. Terrain classification mainly consists of two procedures: feature extraction and classifier training.

Before feature extraction, the image viewpoint is converted from the camera view into a top view for evaluating the region corresponding to the ROI for slope estimation. Inverse perspective transformation is applied for image transformation preprocessing [29].

Visual features are then selected and extracted from the converted RGB image ac-

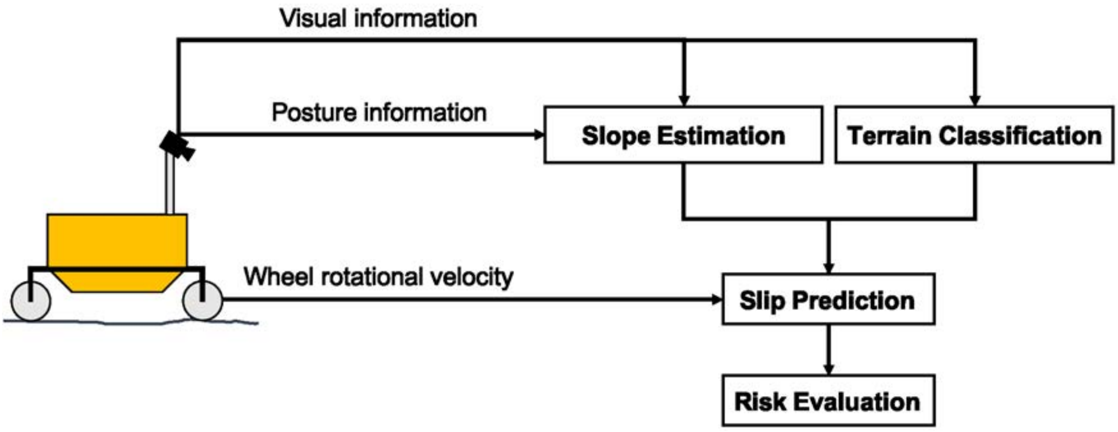


Fig. 2.3: Architecture of proposed slip prediction approach. Input posture and 3D information acquired using the IMU and RGB-D sensor, respectively, are used for slope estimation, and image data is used for terrain classification. Wheel slip is predicted by integrating both the slope estimation and the terrain classification results.

quired using the exteroceptive sensor. In this study, color- and texture-based features are applied for classification. To obtain color-based features, the original RGB image is converted to the L*a*b* color space, where L* represents lightness, and a* and b* represent color dimensions. The mean values of a* and b* inside a patch are calculated to compose a two-element feature vector, and the L* value is not used to reduce the effect of lighting conditions. Texture-based features are also extracted from the obtained images. A texture is a measure of the local spatial variation of image intensity. In this study, energy and contrast, which express the gray-scale distribution homogeneity and image sharpness, respectively, are calculated from the gray level co-occurrence matrix (GLCM). To decompose each image, a superpixel representation is applied instead of fixed-sized patches. Each superpixel agglomerates visually homogeneous pixels while respecting natural scene boundaries [30]. Specifically, simple linear iterative clustering (SLIC), in which a k-means clustering approach is used to efficiently generate superpixels, is used to decompose the image into image patches for feature extraction [31]. Fig. 2.5 shows a sample image of this feature extraction process.

Finally, a classifier is trained using the color- and texture-based features. In this study, the gradient boosting decision tree (GBDT) classifier is used for solving the classification problem. This learning method is selected as the terrain classifier since (1) it can achieve better predictive performance while avoiding overfitting, and (2) color and texture of terrain can be evaluated to improve its classification performance. Gradient boosting is a type of ensemble learning that consists of multiple weak prediction

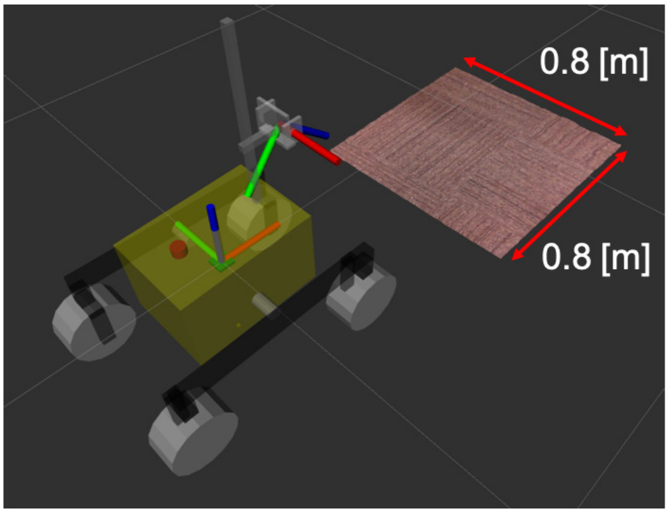
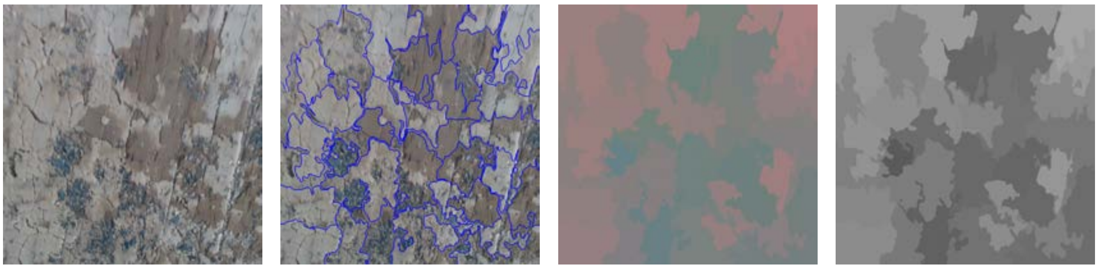


Fig. 2.4: ROI in front of rover in RViz 3D visualization tool. The ROI is expressed by a 3D point cloud of 0.8 m^2. Two coordinate systems are used for the camera and the rover body, respectively, and 3D point cloud information is converted from the camera into the body coordinate system for slope estimation.



(a) (b) (c) (d)

Fig. 2.5: (a) Original visual image of cracked-ground region. (b) Segmented image by superpixels. Blue lines are boundaries for visually homogeneous pixels. (c) Mean color of L*a*b* color space. (d) Mean color of gray-scale image used for texture extraction.

models [32]. In particular, a GBDT comprises multiple decision trees that recursively partition the input space for prediction [33]. Thus, the output of GBDT, $p(c|\mathbf{v})$, is expressed by the posterior probability $p_t(c|\mathbf{v})$ of each decision tree as follows:

$$p(c|\mathbf{v}) = \frac{1}{T}\sum_{t=1}^{T} p_t(c|\mathbf{v}), \tag{2.3}$$

where c, $\mathbf{v}$, and T are the classes, input feature vectors, and number of decision trees, respectively.

2.3.3 Terrain-Dependent Slip Risk Prediction

Definition of Slip Risk

As mentioned in Section 2.1., wheel slip is quantitatively described by the slip ratio. This study classifies the slip ratio as low ($0 < s \leq 0.3$), medium ($0.3 < s \leq 0.6$), and high ($0.6 < s$). In previous literature [34,35], the slip ratio was distinguished into three classes, as in the present study, by referring to the behaviors of a single-wheel testbed with a flight spare wheel for *Curiosity*. Another study noted that wheels exhibit a sharp increase in slip for slip ratio of 0.2–0.3 [36]. Similarly, the experimental results of the present study, as presented later, indicate that the rover travels steadily for slip ratio of up to 0.3 but becomes unstable when the slip ratio reaches 0.6. Based on both previous literature and the rover's mobility in this study, this definition is used as a reasonable indicator for evaluating the wheel slip risk.

Slip Risk Prediction

Before risk prediction, the slip ratio is predicted from a regression curve using the acquired slip versus slope data. The true slip ratio t and a linear basis function model $y(x, \mathbf{w})$, that outputs predicted slip ratio for the given slope angle input x, are expressed as follows:

$$t = y(x, \mathbf{w}) + \epsilon, \tag{2.4}$$

$$y(x, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(x) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(x), \tag{2.5}$$

where ϵ and $\mathbf{w}$ in (2.4) are the additive noise and weight coefficient, respectively, and $\boldsymbol{\phi}(x)$ in (2.5) is the vector of M basis functions. A sum-of-squared error function $E(\mathbf{w})$ is defined to obtain optimized $\mathbf{w}$ with acquired N slope angle inputs and corresponding slip ratios, as follows:

$$\arg\min_{\mathbf{w}} E(\mathbf{w}) = \arg\min_{\mathbf{w}} \frac{1}{2} \sum_{i=1}^{N} \left\{ t_i - \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(x_i) \right\}^2 . \tag{2.6}$$

The slip ratio for the given slope angle is finally predicted by $y(x, \mathbf{w})$. In this study, the Gaussian basis function is used as the basis function to fit nonlinear slip versus slope relation:

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2\sigma^2}\right\}, \tag{2.7}$$

where μ_j is the location of the basis function in the input space and σ, its spatial scale [37].

Then, the slip risk is predicted by comparing the acquired regression to the thresholds mentioned in Section 3.3.1. While the three classes are defined as slip risk common to the terrain, this study redefines the slip risk to a terrain-dependent one by reflecting the slope estimation accuracy. Fig. 2.6 shows the concept of deciding the terrain-dependent threshold for risk prediction. The threshold common to terrain, such as 0.3 and 0.6, is expressed as

$$s = f(\phi). \tag{2.8}$$

To redefine the terrain-dependent threshold in consideration of the slope estimation accuracy, MAE τ is subtracted from ϕ as follows:

$$s' = f(\phi - \tau) = f(\phi'), \tag{2.9}$$

$$MAE : \tau = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{x}_i - x_i\right|, \tag{2.10}$$

where $\hat{x}_i$ and x_i are the estimated slope angle value and ground truth value, respectively. Through the above procedure, the thresholds of each terrain for risk prediction are shifted from common points (ϕ, s) to terrain-dependent ones (ϕ', s'). Note that the regression curve for each terrain has a different inclination because different terrain properties result in specific wheel-terrain interactions. For instance, while wheel slip on homogeneous loose sand monotonously increases as the slope of the traversing surface increases, a rover's behavior on gravel terrain becomes more unpredictable owing to

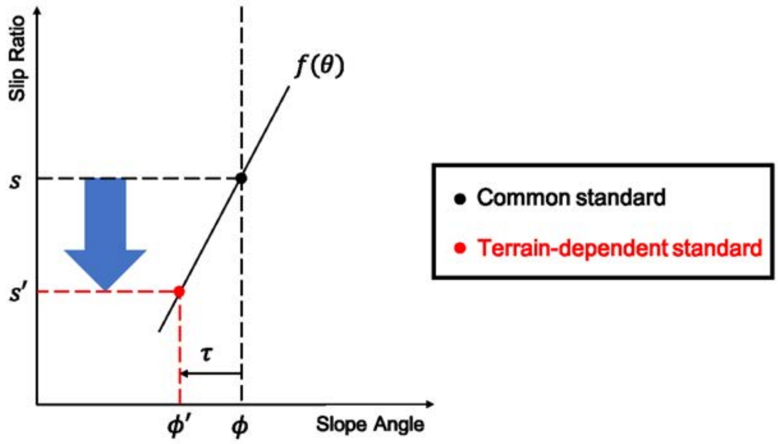


Fig. 2.6: Concept of deciding new threshold considering inclination of function.

the uneven constitution of this terrain. The slip risk is redefined in consideration of the slope estimation accuracy for dealing with the terrain-dependent inclination of the regression curve, especially when a sharp increase in slip ratio results in a difference between the actual and predicted slip risk.

2.4 Experiment

An experiment was performed to validate the proposed approach using datasets including images, 3D point clouds, and slip versus slope data collected with a four-wheeled rover testbed operating on rough terrain. This section describes the testbed, data collection, and experimental conditions in detail.

2.4.1 Testbed

In this study, the El-Dorado IIB four-wheeled rover testbed (Fig. 2.7) is used for data collection. Table 2.1 lists the specifications of the testbed. Each wheel has a driving and steering motor, and the testbed's rocker-type passive suspension mechanism enables getting over obstacles such as small rocks. The testbed is equipped with an Intel®RealSense™D435 RGB-D sensor, an IMU, wheel encoders for each wheel, and potentiometers between the suspension and the main body. The RGB-D sensor simultaneously captures image and depth information with the resolution and field of view (FOV) shown in Table 2.2. Depth information is calculated by stereo imagery and complemented by projecting infrared patterns. The system configuration is im-

Fig. 2.7: El-Dorado IIB rover testbed on sandy terrain. RGB-D sensor, IMU inside the main body, and potentiometers attached to each side of the body are used for acquiring data.

plemented using the Robot Operating System (ROS) and manually steered by using a joystick controller. The controller commands steering motors for turning and driving motors for traveling with angular velocity of at most 0.5 rad/s. When no wheel slip occurs, the testbed can traverse terrain with translational velocity of at most 0.05 m/s.

2.4.2 Data Collection

Data were collected for five terrain types: *Sand*, *Soil*, *Gravel*, *Asphalt*, and *Cracked-ground*. Fig. 2.8 shows examples of these terrain types. Dry, cracked terrain is also classified as *Cracked-ground* because the terrain is solid, unlike *Soil* terrain, and is assumed to be a bedrock region on the surface of Mars. Data for sandy terrain was collected indoors, and data for other terrains were collected outdoors.

Fig. 2.9 (a), (b) shows the indoor test field. A sandbox is uniformly and loosely covered with dry Toyoura Standard Sand; it can be jacked up manually at an inclina-

Table 2.1: Specifications of El-Dorado IIB rover testbed

Size [mm]	L800 $\times$ W650 $\times$ H400
Mass [kg]	23.8
Wheel size [mm]	ϕ 200 $\times$ W100
Wheel base [mm]	600

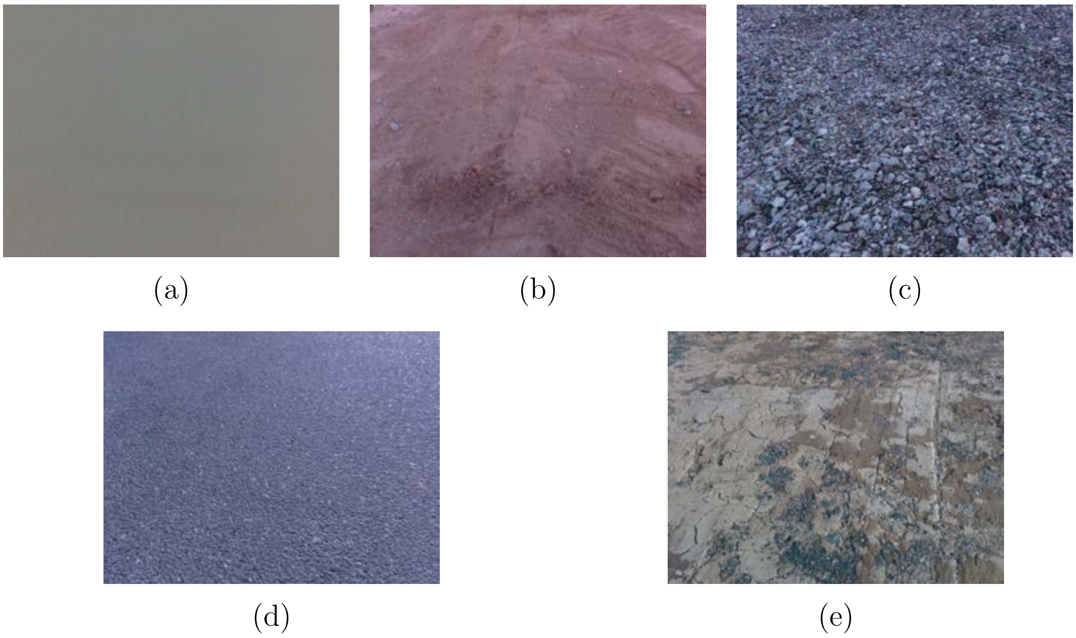


Fig. 2.8: Images of different terrain types ((a) *Sand* in indoor test field and (b) *Soil*, (c) *Gravel*, (d) *Asphalt*, and (e) *Cracked-ground* in outdoor test fields) on which the testbed traversed during data collection. Dry, cracked soil was classified as (e) *Cracked-ground* assuming a bedrock region on the surface of Mars.

tion. The testbed motion was tracked using a motion capture camera with accuracy of 1 mm and recording frequency of 100 fps.

Fig. 2.9 (c), (d) shows the outdoor test fields. For *Soil* and *Gravel* terrains, slopes were made manually. The testbed motion was tracked using a total station with accuracy of 1 mm and recording frequency of 5 fps.

To collect slip versus slope data, the testbed tried to climb the slope for *Sand*, *Soil*, and *Gravel* terrain types. During data collection, image and 3D point cloud data were collected using the RGB-D sensor with recording frequency of 5 fps, posture information was collected by combining data from the IMU and potentiometers, and angular velocity of each wheel was collected from the encoder. The ground truth value

Table 2.2: Specifications of Intel®RealSense™Depth Camera D435 RGB-D sensor

Image resolution [pixel]	320 × 180 – 1920 × 1080
Depth resolution [pixel]	424×240 – 1280 × 720
Frame rate [fps]	15 – 90
Measurement range [m]	0.2 – 10
Field of view of image sensor [°]	H69.4 × V42.5 × D77
Field of view of depth sensor [°]	H85.2 × V58 × D94

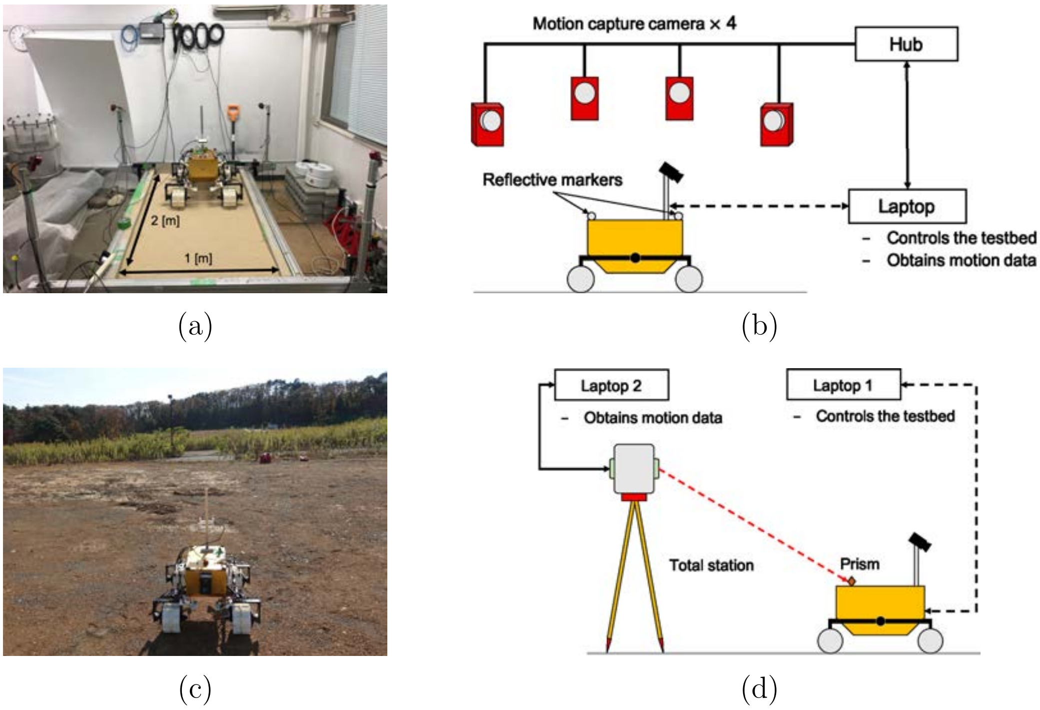


Fig. 2.9: (a) Indoor test field filled with loose, dry sand. (b) Schematic of indoor experimental setup. (c) Outdoor test fields with smooth soil and cracked-ground surfaces. (d) Schematic of outdoor experimental setup.

of the slope angle was obtained from a digital inclinometer. The slip ratio s was calculated as follows:

$$s = 1 - \frac{v}{r\omega} = 1 - \left(\frac{dx}{dt} \cos\theta + \frac{dz}{dt} \sin\theta \right) \frac{1}{r\omega}, \tag{2.11}$$

where v is the translational velocity of the testbed; r, the wheel radius; ω, the angular velocity of the wheel; and θ, the slope angle. Proportional–Integral (PI) control was performed for each wheel to rotate at 0.5 rad/s for traveling at 0.05 m/s.

2.4.3 Experimental Conditions

The experimental conditions for terrain classification and slip risk prediction are described below. The MAE of slope estimation for terrain-dependent slip risk prediction is discussed in Section 4.3.2.

Terrain Classification

The classifier was trained on approximately 500 images to predict the terrain type in 25 test images for verification. Five terrain types—*Sand*, *Soil*, *Gravel*, *Asphalt*, and *Cracked-ground*—are contained in respective superpixels. Manually labeled images were also prepared for images including multiple terrain types. When more than half the pixels in a superpixel belonged to one class, the whole superpixel was classified as that class. During classifier training, stratified K-fold cross-validation with $K = 6$ was applied for evaluating the generalization.

Slip Risk Prediction

The regression curve for each terrain was predicted using the collected slip versus slope data. Before optimizing the function, the hyperparameters of the Gaussian basis function $\phi_j(x)$ were set. The maximum slope angle point of each terrain was regarded as the maximum value of μ_j. Other hyperparameters s and j were decided by grid search. To predict slip risk, absolute errors between the ground truth and the estimated value near $s = 0.3$ and $s = 0.6$ were used as the MAE.

2.5 Results

2.5.1 Terrain Classification Results

Fig. 2.10 shows examples of classification results. Fig. 2.11 shows quantitative results of the terrain classifier as a confusion matrix and a learning curve. By normalizing true positives (TPs), false positives (FPs), true negatives (TNs), and false negatives (FNs), the accuracy (ACC) of each class displayed in the confusion matrix was calculated as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2.12}$$

These results indicate that the classifier predicts reasonable classes for each terrain; however, its ability decreases for *Cracked-ground*, as shown in Fig. 2.11 (a). For instance, for the *Cracked-ground* image in Fig. 2.10, multiple terrain types including *Cracked-ground* and *Soil* with unclear boundaries are present in the image. Therefore,

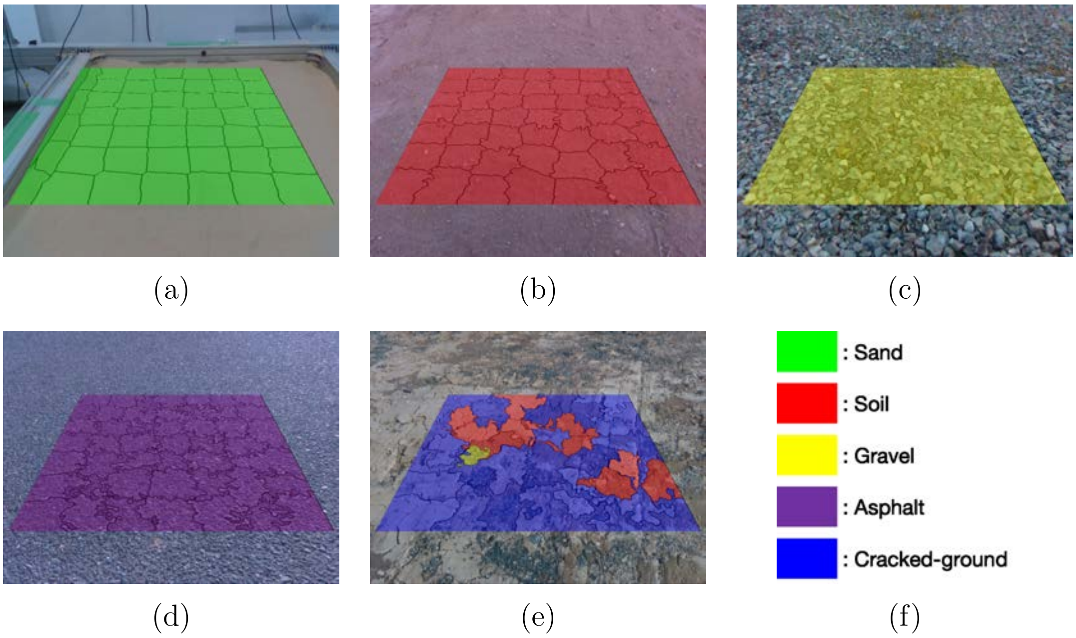
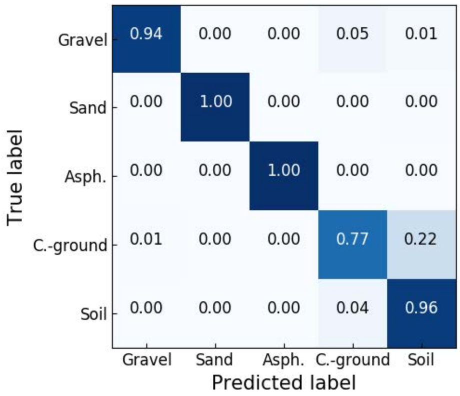


Fig. 2.10: (a), (b), (c), (d), and (e) Examples of classes predicted by terrain classifier for each terrain image. Boundaries in each image indicate the superpixels made by SLIC. (f) Correspondence of classes to color.

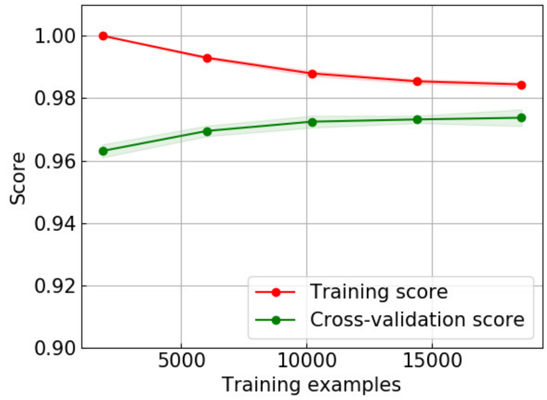
determining classes is difficult even for a human, and the classifier uses these incorrect results to learn to predict terrain types from the image. This incorrect human labeling is the main cause of reduced accuracy. Fig. 2.11 (b) shows the generalization performance of the classifier. Two "learning curves" indicates whether the classifier overfits the training data. The green curve representing the cross-validation score approaches the red curve representing the training score. This trend indicates that iterative training using cross-validation improves the classifier's generalization performance.

Table 2.3: Terrain-independent slip versus slope standards for risk prediction

Terrain	MAE	Slip risk	Slip ratio	Slope angle
Sand	0.75	*low*	$s < 0.26$	$\theta < 4.2$
		medium	$0.26 \leq s < 0.55$	$4.2 \leq \theta < 9.3$
		high	$0.55 \leq s$	$9.3 \leq \theta$
Soil	1.15	*low*	$s < 0.23$	$\theta < 12.5$
		medium	$0.23 \leq s < 0.47$	$12.5 \leq \theta < 15.6$
		high	$0.47 \leq s$	$15.6 \leq \theta$
Gravel	1.15	*low*	$s < 0.25$	$\theta < 16.0$
		medium	$0.25 \leq s < 0.53$	$16.0 \leq \theta < 21.5$
		high	$0.53 \leq s$	$21.5 \leq \theta$



(a)



(b)

Fig. 2.11: Summary of ability and validity of terrain classifier. (a) Confusion matrix of terrain classifier with normalization. (b) Learning curve of terrain classifier. Green and red lines indicate the cross-validation score and training score, respectively. Color bands around each line indicate standard deviation of their scores.

2.5.2 Slip Risk Prediction Results

The slip versus slope relation was acquired using the regression curve and terrain-dependent risk prediction for three terrain types: *Sand*, *Soil*, and *Gravel*. The slip versus slope data obtained during data collection, shown in Fig. 2.12, indicates that wheel slip trends, such as the rate of slip ratio, depend on terrain properties whereas the slip ratio monotonously increases with increasing slope angle for all terrain types. In *Sand* terrain that comprises homogeneous particles, the slip versus slope relation becomes more linear compared to that in other terrain types. While the wheel gains driving force via grousers pushing away particles in front of the wheel, a larger slope angle increases the slip ratio by drawbar pull [18]. In *Soil* terrain, the slip ratio increases sharply when the slope angle exceeds 15° because the fine grain surface suddenly collapses when the slope angle approaches the internal friction angle of the composition. Further, wheel slip in *Soil* terrain is unlikely to occur because the surface is solid until it breaks up. The *Gravel* terrain also makes the rover travel with low wheel slip compared to that in the case of *Sand* terrain; however, the slip versus slope data is irregular owing to the presence of large, nonuniform particles compared to other terrain types [38].

Fig. 2.13 shows a summary of the results for three terrain types. The regression curves for each terrain type fit the monotonic increase of the slip versus slope relation without overfitting the collected data. The thresholds for risk prediction shift from

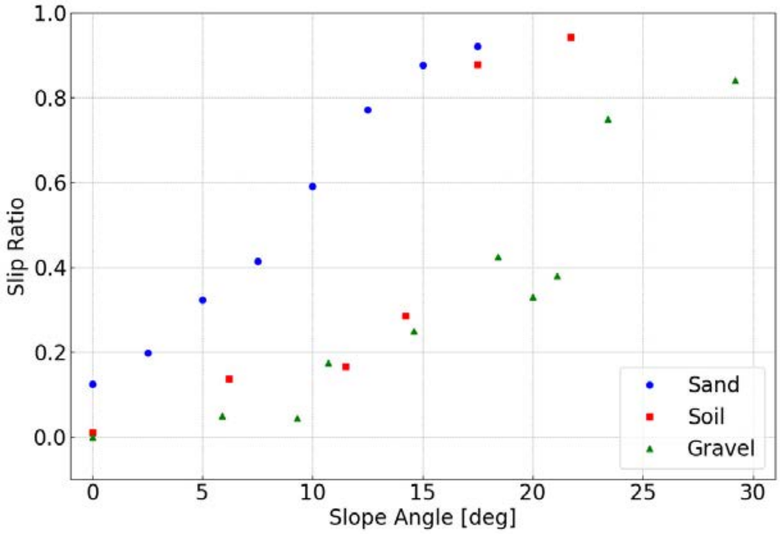
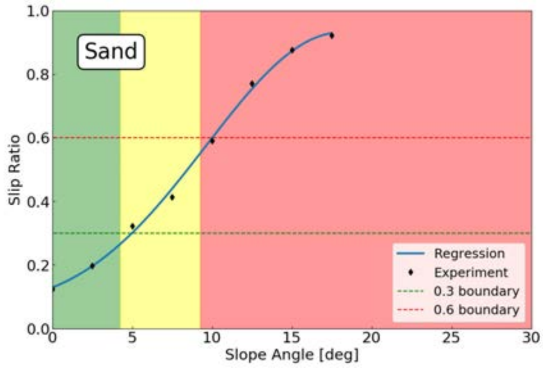


Fig. 2.12: Input slip versus slope data. Blue, red, and green points indicate data acquired on *Sand*, *Soil*, and *Gravel* terrain, respectively.

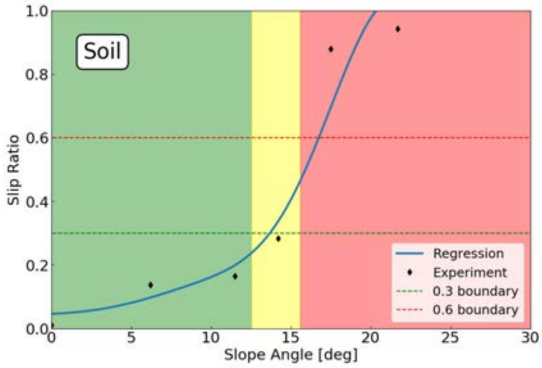
common to terrain-dependent ones using these regression curves, as shown in Table 2.3. Table 2.3 also presents the slope estimation accuracy, described in Section 3.1., as an MAE value. The MAE value for each terrain type shows that slope estimation using the RGB-D sensor can well estimate the slope angle in front of the rover. However, such estimation results can still cause problems for the rover; thus, MAE is subtracted from the slope angle corresponding with common thresholds for risk prediction. For *Sand* terrain, the regression curve linearly increases compared to those for *Soil* and *Gravel* terrains; therefore, the thresholds are not significantly affected by the MAE. When the slope angle reaches 9.3°, the slip risk is considered high, and the rover seems to face difficulties in traversing *Sand* terrain. By contrast, the slope of the regression curve for *Soil* terrain suddenly increases at around 17°; therefore, the slip ratio threshold between medium and high risk decreases to 0.13. This sharp increase causes a sudden transition of the slip risk from low to high. In *Gravel* terrain, the regression curve avoids overfitting the irregularity of the slip versus slope relation. Low slip risk, shown as a green region in Fig. 2.13 (c), indicates that the rover can climb a slope of around 16° safely, whereas it is unable to do so for other terrain types.

2.6 Summary

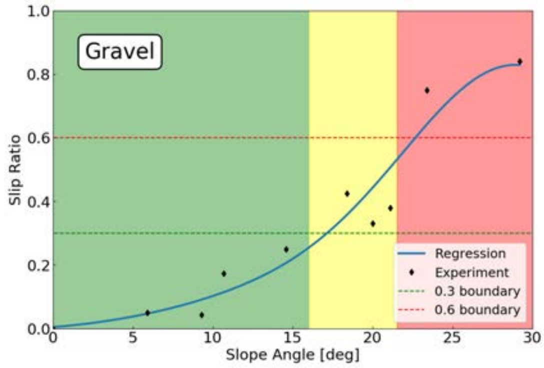
This study proposes a data-driven approach to predict the slip risk of a rover operating on rough terrain by estimating the slope angle and classifying surrounding



(a)



(b)



(c)

Fig. 2.13: (a), (b), and (c) Slip risk prediction result for three terrain types: *Sand*, *Soil*, and *Gravel*. Black points indicate input data obtained from data collection. The blue line indicates the predicted regression curve. Green, yellow, and red regions indicate low, medium, and high slip risks, respectively. Green and red dotted lines indicate the common boundaries of low and medium slip and medium and high slip, respectively.

terrain using visual information. Terrain-dependent slip risk is predicted by setting the slip-risk criterion for each terrain type in consideration of slope estimation accuracy. Experimental validation is performed using datasets acquired from a four-wheeled rover testbed traversing different types of rough terrains. The proposed approach can (1) estimate the slope angle within an MAE of 1.15°, (2) classify five terrain types with at least 77% accuracy, and (3) predict the wheel slip risk on three terrain types in consideration of the slope estimation accuracy, which may be affected by noises such as those contained in 3D information. The contribution of this study is to predict future wheel slip by interpreting terrain properties using a vision-based terrain classifier. In order to incorporate terrain classification with geometric assessment, terrain-dependent thresholds for the slip versus slope relation are introduced.

Future studies will implement online learning to update slip versus slope relation. This can be useful especially in early exploration stages to adapt newly found terrain, for changing terrain conditions such as compaction level, or in case of wheel-terrain relation changes due to the wheels' damage. Another direction is to further extend the current method to predict wheel slip for handling mixed terrain. A more detailed evaluation method will be implemented in such a scenario, such as slip risk prediction considering wheel configuration.

Chapter 3

Active Domain Adaptation via Robotic Information Gathering

3.1 Introduction

Improved autonomy of planetary rovers is crucial to exploring vast surfaces on celestial bodies within demanding mission timelines. For instance, the NASA Mars Sample Return campaign will require the sample fetch rover to traverse up to 20 km within 150 Martian solar days to collect materials (e.g., rocks, soils, and atmosphere) [39]. Although past Martian rovers have possessed autonomous mobility to maintain safety and efficiency [15], fast traverse rates were not possible due to the frequent human intervention required to assess hazardous terrain amid non-negligible communication latency. Therefore, self-reliant terrain assessment is key to implementing faster and more extended rover travel.

Deformable surfaces covering extraterrestrial terrain are particularly hazardous for rovers: excessive wheel slip degrades driving speed, increases energy consumption, and eventually causes permanent entrapment in loose, granular sand. For example, the Curiosity rover in the Mars Science Laboratory mission experienced a significant slip at Hidden Valley on terrain comprising rippled sand and was forced to divert to a conservative, redundant route [14]. Similarly, the Opportunity rover in the Mars Exploration Rover mission faced wheel sinkage situations at Endeavour Crater [40]. Although rovers must assess such underlying hazards prior to execution, predicting traversability on unexplored terrain is challenging owing to its stochastic nature arising from factors including terrain geometry, physical properties, and surface condition [19]. To remotely

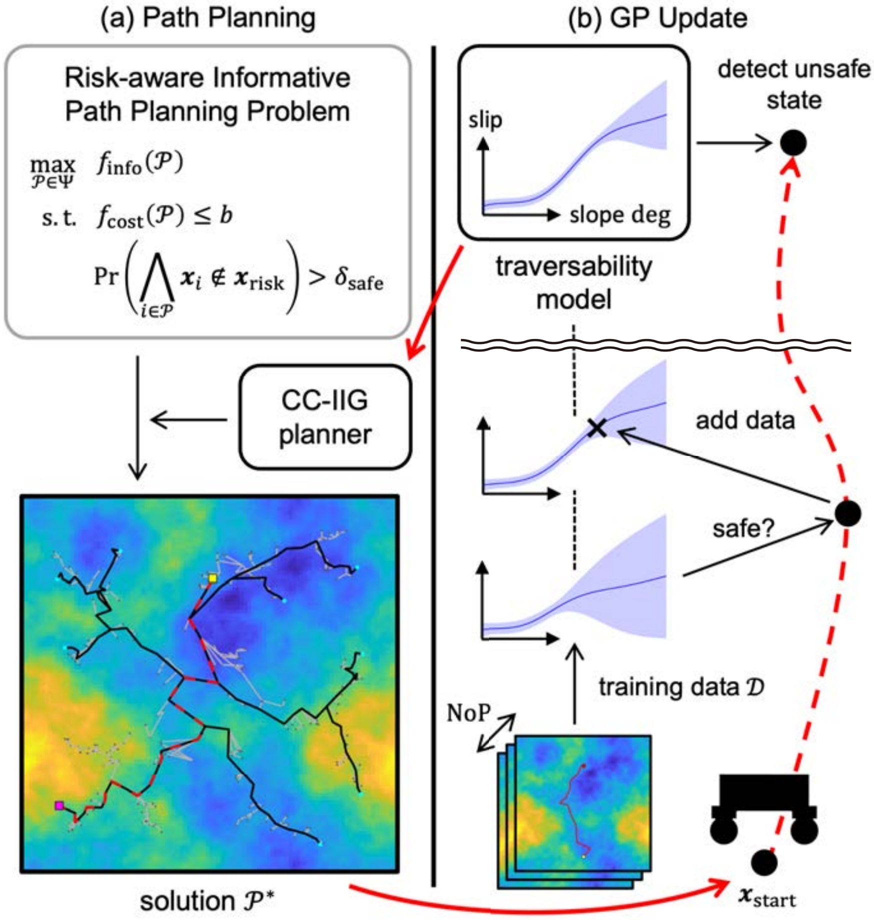


Fig. 3.1: Active traversability learning framework concept. A CC-IIG algorithm plans an informative trajectory (left) and the rover measures traverse data to update GPs while assessing immobilized risk (right). Red arrows illustrate sequential planning and learning processes. NoP is the number of paths traveled so far.

estimate future immobilized risk without accessing non-geometric information, it is vital to assess dependencies between observable surface geometry and traverse experience gathered by *in situ* measurements.

In this chapter, we devise an active traversability learning framework that sequentially gathers *in situ* traverse data to estimate a latent traversability model for unexplored environments. The challenge in this work is that a rover must experience *a priori* unknown hazards without facing extreme slip situations. Thus, as shown in Fig. 3.1, we propose an iterative two-stage framework, which employs an informative path planning (IPP) algorithm and takes *in situ* measurements to refine the model estimation. A Gaussian process (GP) is applied to learn traversability by building a spatial representation from remotely measured terrain geometry. Our strategy first searches for informative regions to reduce uncertainty by employing a sampling-based motion planning algorithm that efficiently finds non-myopic solutions in continuous configuration space. We design a chance-constrained incrementally-exploring information gathering (CC-IIG) algorithm, which introduces a chance constraint formulation to the IIG algorithm proposed in [41]. Given a predictive distribution in GPs, CC-IIG can evaluate stochastic immobilized risk with a specified confidence level. Further, CC-IIG equips a fault-tolerant risk inference that assumes future observations to update the predictive distribution for improving reachability toward informative places. The framework then takes *in situ* measurements along the route to obtain the spatial distribution. A replanning operation is implemented to avoid inadmissible immobilized risk, using the newest traversability estimation together with its confidence. As GP estimates reduce uncertainty, the framework can acquire informative yet hazardous traverse data along feasible trajectories by exploiting learned traversability described as the predictive distribution.

The contributions of this chapter are as follows:

- We formulate an active traversability learning framework that sequentially gathers *in situ* traverse data to update GPs while assessing rover immobilized risk.
- We design CC-IIG, a sampling-based informative path planning algorithm coupled with a fault-tolerant risk inference.
- Simulation studies in rough terrain environments validate the proposed framework learning latent traversability while considering rover safety.

To the best of our knowledge, this work is the first approach that exploits an IPP algorithm to actively learn rover traversability.

The remainder of the chapter is organized as follows. In Section 3.2, the related works are examined. Section 3.3 introduces the components of the problem and formalizes the risk-aware IPP problem. Section 3.4 presents the active traversability learning framework. Section 3.5 details the proposed framework through simulations. Finally, Section 3.6 concludes this chapter.

3.2 Related Work

This work integrates the notion of traversability prediction, informative path planning, and planning under uncertainty. The following subsections summarize the related literature in these fields.

3.2.1 Traversability Prediction

Numerous studies have tackled rover traversability estimation using *in situ* measurements. Example applications include vision-based terrain classification [42, 43], self-supervised approaches combining exteroceptive (e.g., vision) and proprioceptive (e.g., vibration) sensing [21,22], data-driven slip prediction via regression algorithms [1, 27,44], and navigation strategy based on online cost map updates [45]. However, these approaches could not answer a question: how can we predict unsafe traversability for unknown states? As rovers typically prioritize safe travel even when this results in overly conservative routes, rovers struggle with insufficient measurements on hazardous terrain. One approach in [46] alleviates this issue using other terrain slip properties via transfer learning. However, this approach still fails for terrain cases where no slip property data has been acquired previously.

3.2.2 Informative Path Planning

To improve the quality of information metrics, robots are required to efficiently obtain *in situ* measurements subject to given constraints, such as time or energy. This problem can be viewed as IPP, i.e. finding a trajectory that maximizes information quality. Previous work solved IPP problems in discrete domains to model unknown cost fields [47] or environmental phenomena [48–50], but computational complexity involved meant the results were sensitive to the configuration space size. Therefore, for continuous domain problems, sampling-based planning strategies, such as rapidly-exploring random tree (RRT) [51] and its asymptotically optimal variant (RRT*) [52], are exploited to search for informative paths. Rapidly-exploring information gathering (RIG) algorithms designed in [53] are non-myopic robotic exploration techniques with asymptotic optimality. Incrementally-exploring information gathering (IIG) [41] extends RIG for sequential motion planning and execution by proving when to stop the optimization procedure. Similarly, [54, 55] both propose adaptive strategies for iterative inference of spatial distribution for search tasks, while their latent processes do not threaten agents.

With applications to planetary exploration, [56–59] present information-theoretic decision-making algorithms. [56,57] seek trajectories to enrich scientific knowledge (e.g., distribution of rocks or minerals), whereas [58, 59] derive texture-rich paths to improve visual odometry performance. While no research aspired to actively observe hazardous events, assessing traversability should be a natural mechanism to enhance rover autonomy for planetary surface missions.

3.2.3 Planning Under Uncertainty

Planning under uncertainty requires the handling of probabilistic risks to achieve safe and reliable robotic navigation. A chance constraint formulation can guarantee feasible solutions by transforming stochastic constraints into deterministic ones within a prescribed tolerance. For motion planning problems, sampling-based algorithms in [60–62] employ chance constraints to generate collision-free trajectories in high-dimensional configuration space. Rapidly-exploring random belief tree in [61] and chance-constrained RRT* (CC-RRT*) in [62] promise robust and asymptotically optimal solutions by combining the chance-constrained approach and the RRT* framework. This idea is then applied to off-road navigation, including extraterrestrial exploration. A planning algorithm in [63] applies CC-RRT* to navigate a tracked vehicle along a robust trajectory for uncertain slip while focusing on the collision chance rather than immobility. Another algorithm in [64] explicitly checks traversability over inclined terrain using reasonably converged regression models, though an ill-converged spatial distribution would degrade its performance. An information-gathering technique in [65] uses the chance-constrained approach to map the energy costs of ground vehicles on uncertain terrain. Although this shares our motivation of simultaneously learning and assessing spatial processes, using energy consumption as a planning objective would not directly assess vehicle safety.

3.3 Problem Formulation

This section describes the quantitative expressions for traversability, the Gaussian processes for learning its spatial distribution, and the risk-aware IPP problem.

3.3.1 Gaussian Processes for Traversability Modeling

Wheeled rovers can experience longitudinal and lateral slips when traversing through rough terrain. The slip ratios s_x and s_y quantitatively describe rover traversability in each direction as follows:

$$s_x = \begin{cases} \left(v_{\mathrm{ref}} - v_x\right) / v_{\mathrm{ref}}, & \text{if } v_x \leq v_{\mathrm{ref}}, \\ \left(v_{\mathrm{ref}} - v_x\right) / v_x, & \text{otherwise}, \end{cases} \tag{3.1}$$

$$s_y = v_y / v_{\mathrm{ref}}, \tag{3.2}$$

where v_x and v_y are the actual longitudinal and lateral velocities, and v_{ref} is the reference velocity [19]. A positive s_x represents a traverse slower than commanded, with $s_x = 1$ representing complete rover entrapment in rough terrain.

We intend to predict the traversability in unvisited regions. Terrain geometric information is a reasonable choice as a function input of spatial distribution because 1) onboard rover sensors can assess terrain surface and 2) terrain geometry strongly correlates to rover slip behavior, as described in [18]. The geometric information is defined as rover pitch θ_x and roll θ_y dominating s_x and s_y, respectively, as shown in Fig. 3.2.

GPs are non-parametric Bayesian regression approaches employing statistical inference to learn dependencies between points in a dataset [66]. We use GPs to model the rover traversability because 1) they can incorporate a set of undefined terrain parameters into uncertainty, 2) the magnitude of uncertainty indicates where rovers should take *in situ* measurements, and 3) they can handle nonlinearity in rover-terrain interactions. We assume that an actual latent function $f_\bullet$ casts the arbitrary terrain geometry $\theta_\bullet$ to traversability $s_\bullet$ in each direction as follows:

$$s_x = f_x(\theta_x) + \epsilon_x, \tag{3.3}$$

$$s_y = f_y(\theta_y) + \epsilon_y, \tag{3.4}$$

where $\epsilon_\bullet$ denotes a zero-mean normally distributed error term with variance $\sigma_\bullet^2$. Here, $f_\bullet$ follows the multivariate Gaussian distribution defined by the mean function $m(\theta_\bullet)$ and positive definite covariance function $k(\theta_\bullet, \theta'_\bullet)$. Given a training set $\mathcal{D}_\bullet = \{(\theta_{\bullet,i}, s_{\bullet,i}) | i = 1, \ldots, n\}$, the traversability can be probabilistically modeled as

$$f^*_\bullet | \theta^*_\bullet, \mathcal{D}_\bullet \sim \mathcal{N}(m(\theta^*_\bullet), v(\theta^*_\bullet)), \tag{3.5}$$

where m and v express the predictive mean and variance described by

$$m(\theta^*_\bullet) = \mathbb{E}\left[f^*_\bullet\right] = \boldsymbol{k}_*^{\mathrm{T}} \left[\boldsymbol{K} + \sigma^2 \boldsymbol{I}\right]^{-1} \boldsymbol{s}_\bullet, \tag{3.6}$$

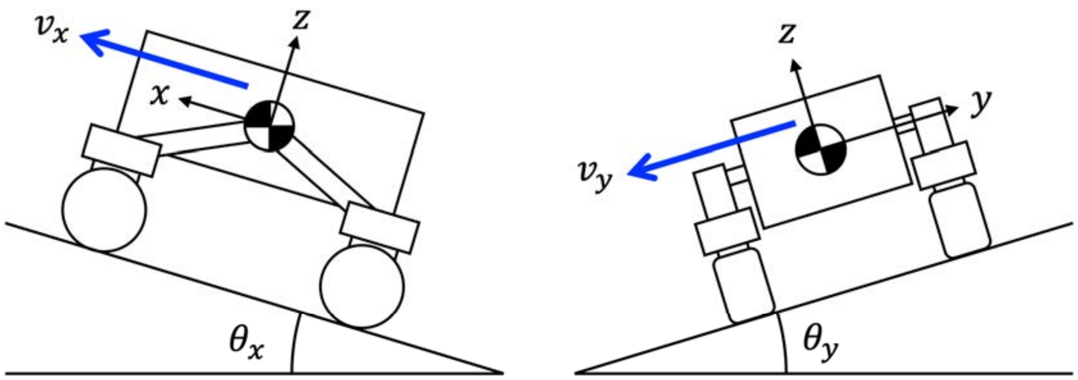


Fig. 3.2: Terrain inclinations and corresponding rover velocities in longitudinal (left) and lateral (right) directions.

$$v(\theta^*_\bullet) = \mathbb{V}\left[f^*_\bullet\right] = k_{**} - \boldsymbol{k}_*^{\mathrm{T}} \left[\boldsymbol{K} + \sigma^2 \boldsymbol{I}\right]^{-1} \boldsymbol{k}_*, \tag{3.7}$$

where $\boldsymbol{K} = \boldsymbol{K}(\boldsymbol{\theta}_\bullet, \boldsymbol{\theta}_\bullet)$, $\boldsymbol{k}_* = \boldsymbol{k}(\boldsymbol{\theta}_\bullet, \theta^*_\bullet)$, $k_{**} = k(\theta^*_\bullet, \theta^*_\bullet)$ with n training inputs $\boldsymbol{\theta}_\bullet$ and a test input $\theta^*_\bullet$, and $\boldsymbol{s}_\bullet$ and $\boldsymbol{I}$ are the measurements and the $n \times n$ identity matrix, respectively.

Note that for the covariance function, we combine the squared exponential and linear kernels as follows:

$$k(x, x') = \sigma^2_{f,s} \exp\left(-\frac{|x - x'|^2}{2l^2}\right) + \sigma^2_{f,l} x x', \tag{3.8}$$

where $\sigma^2_{f,s}$ and $\sigma_{f,l}$ are the signal standard deviations, and l is the characteristic length scale. The linear kernel is helpful for efficiently learning the traversability trend with increasing slope gradient.

3.3.2 Risk-aware Informative Path Planning

Let $\mathcal{P} = \{\boldsymbol{x}_i | i = 0, \ldots, n\}$ be a sequence of rover states starting from $\boldsymbol{x}_0 = \boldsymbol{x}_{\text{start}}$. The node contains positional information of the rover in 3D space, defined as $\boldsymbol{x} = (x, y, z)$. We formally describe the risk-aware IPP problem as follows:

$$\max_{\mathcal{P} \in \Psi} \quad f_{\text{info}}(\mathcal{P}) \tag{3.9a}$$

$$\text{s.t.} \quad f_{\text{cost}}(\mathcal{P}) \leq b \tag{3.9b}$$

$$\Pr\left(\bigwedge_{i \in \mathcal{P}} \boldsymbol{x}_i \notin \boldsymbol{x}_{\text{risk}}\right) > \delta_{\text{safe}} \tag{3.9c}$$

where Ψ is the space of possible trajectories, f_{info} is the function that returns the information quality along $\mathcal{P}$, b is the available budget, and f_{cost} is the function that calculates the cost associated with $\mathcal{P}$. In (3.9c), $\boldsymbol{x}_{\text{risk}}$ denotes an arbitrary hazardous state and δ_{safe} is the user-defined confidence level of safety. The condition in (3.9c) represents that all rover states in $\mathcal{P}$ satisfy the given constraints with a greater likelihood of δ_{safe}. In contrast to the conventional path planning problems, there is no predefined goal state in this problem [67].

We provide the specific mathematical notations for formulating the traversability learning as a risk-aware IPP problem. We elaborate traversability models in the longitudinal and lateral directions using two independent GPs; each GP input is given as $\theta_{x,i} = \theta_x(\boldsymbol{x}_{i-1}, \boldsymbol{x}_i)$ or $\theta_{y,i} = \theta_y(\boldsymbol{x}_{i-1}, \boldsymbol{x}_i)$ by connecting two consecutive states. For the objective function in (3.9a), we designed the mean entropy (ME) formula along $\mathcal{P}$ as follows:

$$f_{\text{info}}\left(\mathcal{P}\right) = \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \left(H(\theta_{x,i}) + H(\theta_{y,i})\right), \tag{3.10}$$

where $|\mathcal{P}|$ is the number of states in $\mathcal{P}$, and H denotes information gain in each direction, given by:

$$H(\theta_{\bullet,i}) = \frac{1}{2} \log 2\pi e v(\theta_{\bullet,i}), \tag{3.11}$$

where v is the predictive variance in (3.7) with substituted $\boldsymbol{\theta}_{\bullet} = [\theta_{\bullet,1}, \ldots, \theta_{\bullet,i-1}]^{\mathrm{T}}$ and $\theta^* = \theta_{\bullet,i}$. In general, (3.10) indicates that higher information quality contributes to lower GP uncertainties. Other metrics such as mutual information can be applied to the objective function [48], although they require more computational resources than ME does.

We use the Euclidean distance metric as the cost function in (3.9b),

$$f_{\text{cost}}\left(\mathcal{P}\right) = \sum_{i \in \mathcal{P}} \|\boldsymbol{x}_{i-1} - \boldsymbol{x}_i\|. \tag{3.12}$$

Other metrics, such as time or fuel, can be easily applied until the cost function is strictly positive and finite [67].

Finally, we formulate chance constraints to translate stochastic constraints into deterministic ones for evaluating probabilistic risk in (3.9c). We assess immobilized risk by considering whether the slips s_x and s_y observed in the timestep $\boldsymbol{x}_{i-1}$ to $\boldsymbol{x}_i$ exceeds the predefined maximally allowable value $s_{\bullet}^{\text{th}}$ in each direction. The arbitrary risk function in (3.9c) is therefore arranged as the joint chance constraint

$$\Pr\left(\bigwedge_{i\in\mathcal{P}} s_x(\theta_{x,i}) < s_x^{\text{th}} \wedge s_y(\theta_{y,i}) < s_y^{\text{th}}\right) > \delta_{\text{safe}}. \tag{3.13}$$

The constraint (3.13) probabilistically guarantees that the slip will not violate $s_\bullet^{\text{th}}$ in both directions for the entire rover states given by $\mathcal{P}$. We then decompose (3.13) to resolve computational intractability by complementing joint events with De Morgan's law and applying Boole's inequality [62] as follows:

$$\sum_{i\in\mathcal{P}} \Pr\left(s_x(\theta_{x,i}) \geq s_x^{\text{th}} \vee s_y(\theta_{y,i}) \geq s_y^{\text{th}}\right) \leq 1 - \delta_{\text{safe}}. \tag{3.14}$$

Given the GP with any training set $\mathcal{D}'_\bullet = \{(\theta_{\bullet,j}, s_{\bullet,j}) | j = 1, \dots, m\}$, we can calculate a deterministic constraint $\kappa_{\bullet,i}$ in each direction using

$$\begin{aligned} \Pr\left(s(\theta_{\bullet,i}) \geq s_\bullet^{\text{th}} | \mathcal{D}'_\bullet\right) &= 1 - \Phi\left(\frac{s_\bullet^{th} - m(\theta_{\bullet,i})}{\sqrt{v(\theta_{\bullet,i})}}\right), \\ &= \kappa_{\bullet,i}, \end{aligned} \tag{3.15}$$

where Φ is the cumulative distribution function of the standard normal distribution. As (3.14) is expressed with a union set of constraints in both directions, we can finally reformulate (3.13) by adding the deterministic values together as follows:

$$\sum_{i\in\mathcal{P}} (\kappa_{x,i} + \kappa_{y,i}) \leq 1 - \delta_{\text{safe}}. \tag{3.16}$$

3.4 Active Traversability Learning

This section presents the active traversability learning framework, which sequentially solves the risk-aware IPP problem and updates the GPs with gathered *in situ* traverse data. Here, unlike active learning in ML [68] where models query for labels, the term "active" in this framework refers to the ability of the framework to autonomously explore and collect traverse data, in contrast to conventional passive methods that rely on paths generated for other mission objectives, such as point-to-goal navigation. We first formulate the synthesis of the path planning and GP update stages in Algorithm 1, then specify our designed sampling-based informative path planning method in Algorithm 2. The symbols x and y are omitted throughout the algorithm description for brevity. This simplification is also applied to θ, s, $\mathcal{D}$, and $\mathcal{GP}$. For instance, θ has pitch and roll direction values as $\theta.x = \theta_x$ and $\theta.y = \theta_y$, respectively. We assume the given environment $\mathcal{M}$ is covered with homogeneous material. We also provide a small safe region θ^{safe} in each direction that ensures rovers never experience immobilized situations within given thresholds. This assumption is reasonable as existing planetary exploration missions have selected benign terrain for landing sites to ensure safe rover operation besides scientific objectives [69].

3.4.1 Traversability Learning (Algorithm 1)

We integrate two stages in Algorithm 1: path planning to find informative locations (lines 3–4) and GP updates via *in situ* measurements (lines 5–8). For the planning process, CC-IIG is used to extend the IIG algorithm in [41] with chance constraint formulation. CC-IIG generates aggressive motion toward informative regions by employing fault-tolerant risk inference. On the other hand, the GP updates process cautiously checks the feasibility of given trajectories to avert immobilized risk. Through GP estimates reduce uncertainty, the algorithm can attain informative yet hazardous traverse data along feasible trajectories. The following subsections detail the procedures of Algorithm 1.

Initialization

In line 1, the algorithm executes an initial exploration to find an informative path with ending state $\boldsymbol{x}$. CC-IIG deterministically avoids steep terrain above θ^{safe} to obtain *in situ* measurements for all states. After the first iteration, the algorithm presents the learned traversability model $\mathcal{GP}$, yet its performance is biased towards gentle terrain. Beyond such areas, CC-IIG then explores more informative, hazardous processes.

Risk-aware GP Update

In lines 3–4, CC-IIG produces $\mathcal{P}_I$ beginning from the previous ending state to the next terminal $\boldsymbol{x}_{\text{target}}$. CC-IIG allows aggressive maneuver toward hazardous regions by assuming future observations via GP estimates to temporarily update the predictive distribution. Then, in lines 5–8, the `isProbFeasible` function checks the immobilized risk at $\boldsymbol{x}_i$ (one step ahead of the current state), using ongoing estimates $\mathcal{GP}$ as follows:

$$\kappa_{x,i} + \kappa_{y,i} \leq 1 - \delta_{\text{safe}}. \tag{3.17}$$

Algorithm 1 Active Traversability Learning

Input: initial state $\boldsymbol{x}_{\text{start}}$, environment $\mathcal{M}$, safe region θ^{safe}
Output: learned traversability model $\mathcal{GP}$

$\mathcal{GP}, \boldsymbol{x} \leftarrow \texttt{CC}-\texttt{IIG.init}(\boldsymbol{x}_{\text{start}}, \mathcal{M}, \theta^{\text{safe}})$
while $\texttt{Information}(\boldsymbol{x}) \geq \delta_{\text{ME}}$ **do**
  $\boldsymbol{x}'_{\text{start}} \leftarrow \boldsymbol{x}$
  $\mathcal{P}_I, \boldsymbol{x}_{\text{target}} \leftarrow \texttt{CC}-\texttt{IIG}(\boldsymbol{x}'_{\text{start}}, \mathcal{M}, \mathcal{GP})$
  while $\boldsymbol{x} \notin \boldsymbol{x}_{\text{target}}$ or $\texttt{isProbFeasible}(\boldsymbol{x}, \mathcal{GP})$ **do**
    $\boldsymbol{x}, \theta, s \leftarrow \texttt{MoveTo}(\mathcal{P}_I), \mathcal{GP}.\mathcal{D} \leftarrow \cup \{(\theta, s)\}$
    $\mathcal{GP} \leftarrow \texttt{UpdateGP}()$
  end while
end while
return $\mathcal{GP}$

Here, (3.17) calculates the slip probability between $\boldsymbol{x}_{i-1}$ and $\boldsymbol{x}_i$ as insignificant to s^{th} in each direction. Considering uncertainty as shown in (3.15) enables the feasibility of state transition to be guaranteed within the given confidence level δ_{safe}, even for high variance GPs such as in the early exploration phase. The GPs are updated with newly measured data until reaching $\boldsymbol{x}_{\text{target}}$ if the consecutive states are reachable. Otherwise, the replanning operation is triggered to detour locations where current GP estimates cannot assure feasible traverse. After updating the GPs, the `Information` function calculates ME using (3.10) from $\boldsymbol{x}_{\text{start}}$ to check convergence of the gathered traverse information. The framework continues the above procedures until ME reaches a user-specified threshold value δ_{ME} to eventually output the learned traversability model $\mathcal{GP}$.

3.4.2 Path Planning (Algorithm 2)

The designed CC-IIG algorithm is used to solve the risk-aware IPP problem with given GPs. IIG is used as the baseline because 1) it can find non-myopic information-rich trajectories in continuous space, and 2) it naturally terminates the optimization procedure for subsequent tasks, including navigation and iterative planning, by checking the convergence of information metrics. Algorithm 2 shows complete procedures for CC-IIG. We first describe the process shared with IIG then specify the extensions added.

The algorithm initializes a tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ in lines 1–4. The subsequent main loop starts with sampling a point in the environment $\mathcal{M}$ to generate a new node $\boldsymbol{x}_{\text{feasible}}$ from its nearest neighbor in the tree (lines 6–9). All nearby nodes around $\boldsymbol{x}_{\text{feasible}}$ are extracted in line 10. Then each node is recast into a new node $\boldsymbol{x}_{\text{new}}$ by the `Steer` function in line 12, considering rover motion constraints. If the tree can grow towards $\boldsymbol{x}_{\text{new}}$ free from hazards, the algorithm evaluates the information quality and cost in lines 16–18. In lines 19–26, the algorithm adds $\boldsymbol{x}_{\text{new}}$ to the tree only when it meets a partial ordering condition [53], and the resulting cost does not surpass the given budget. After completing tree expansion, the algorithm finally selects the most informative path with its ending state using the `PathSelection` function (line 32). Without a specified goal position, CC-IIG must decide the trajectory to execute from multiple feasible trajectories.

Fault-tolerant Risk Inference

Our extension in lines 13-15 is added for awareness regarding immobilized risk using given GPs while increasing the probability of visiting informative locations. In line 13, a dataset $\hat{\mathcal{D}}$ is created by assuming slips from the initial position using the given GPs and treating these as 'ground truth' information. With the GPs trained by the augmented dataset, the `isProbFeasible` function checks the chance constraint violation at $\boldsymbol{x}_{\text{new}}$ by calculating the joint probability using (3.16). The motivation behind the data augmentation is to alleviate the issue of using outdated information to evaluate reachability. By pretending to obtain *in situ* measurements, CC-IIG can employ more tightened GPs to improve route intersection with informative regions. Although the fault-tolerant risk inference would force rovers into daunting situations, Algorithm 1 can later strictly assess immobilized risk with the refined GPs fed with obtained measurements. Furthermore, $\hat{\mathcal{D}}$ converges to the factual information as the given GPs reduce uncertainty via iterative planning, enabling CC-IIG to produce informative and safe route.

Algorithm 2 CC-IIG

Input: initial state $\boldsymbol{x}_{\text{start}}$, environment $\mathcal{M}$, given GPs $\mathcal{GP}$
Output: most informative path $\mathcal{P}_I$, ending state $\boldsymbol{x}_{\text{target}}$

$I_{\text{init}} \leftarrow \texttt{Information}(\boldsymbol{x}_{\text{start}}), C_{\text{init}} \leftarrow 0$
$n \leftarrow \langle \boldsymbol{x}_{\text{start}}, C_{\text{init}}, I_{\text{init}} \rangle$
$\mathcal{V} \leftarrow \{n\}, \mathcal{V}_{\text{closed}} \leftarrow \emptyset, \mathcal{E} \leftarrow \emptyset, I_{\text{RIC}} \leftarrow \emptyset$
$n_{\text{sample}} \leftarrow 0$
while $\texttt{GradientRIC}(I_{\text{RIC}}) < 0$ **do**
  $\boldsymbol{x}_{\text{sample}} \leftarrow \texttt{Sample}(\mathcal{M})$
  $n_{\text{sample}} \leftarrow n_{\text{sample}} + 1$
  $\boldsymbol{x}_{\text{nearest}} \leftarrow \texttt{Nearest}(\boldsymbol{x}_{\text{sample}}, \mathcal{V} \backslash \mathcal{V}_{\text{closed}})$
  $\boldsymbol{x}_{\text{feasible}} \leftarrow \texttt{Steer}(\boldsymbol{x}_{\text{nearest}}, \boldsymbol{x}_{\text{sample}}, \Delta)$
  $\mathcal{V}_{\text{near}} \leftarrow \texttt{Near}(\boldsymbol{x}_{\text{feasible}}, \mathcal{V} \backslash \mathcal{V}_{\text{closed}}, r)$
  for all $n_{\text{near}} \in \mathcal{V}_{\text{near}}$ **do**
    $\boldsymbol{x}_{\text{new}} \leftarrow \texttt{Steer}(\boldsymbol{x}_{\text{near}}, \boldsymbol{x}_{\text{feasible}}, \Delta)$
    $\hat{\mathcal{D}} \leftarrow \texttt{EstimateTransition}(\boldsymbol{x}_{\text{near}}, \mathcal{GP})$
    $\mathcal{GP}' \leftarrow \texttt{UpdateGP}(\cup \{\hat{\mathcal{D}}\})$
    if $\texttt{isProbFeasible}(\boldsymbol{x}_{\text{new}}, \mathcal{GP}')$ **then**
      $I_{\text{new}} \leftarrow I_{\text{near}} + \texttt{Information}(\boldsymbol{x}_{\text{new}})$
      $C_{\text{new}} \leftarrow C_{\text{near}} + \texttt{Cost}(\boldsymbol{x}_{\text{near}}, \boldsymbol{x}_{\text{new}})$
      $n_{\text{new}} \leftarrow \langle \boldsymbol{x}_{\text{new}}, C_{\text{new}}, I_{\text{new}} \rangle$
      if $\texttt{Prune}(n_{\text{new}})$ **then**
        Delete n_{new}
      else
        $I_{\text{RIC}} \leftarrow \texttt{append}(I_{\text{RIC}}, (\frac{I_{\text{new}}}{I_{\text{near}}} - 1)/n_{\text{sample}})$
        $n_{\text{sample}} \leftarrow 0$
        $\mathcal{E} \leftarrow \cup \{(n_{\text{near}}, n_{\text{new}})\}, \mathcal{V} \leftarrow \cup \{n_{\text{new}}\}$
        if $C_{\text{new}} > b$ **then**
          $\mathcal{V}_{\text{closed}} \leftarrow \mathcal{V}_{\text{closed}} \cup \{n_{\text{new}}\}$
        end if
      end if
    end if
  end for
end while
$\mathcal{P}_I, \boldsymbol{x}_{\text{target}} \leftarrow \texttt{PathSelection}(\mathcal{V}, \mathcal{E})$
return $\mathcal{P}_I, \boldsymbol{x}_{\text{target}}$

Automatic Termination Condition

In [41], IIG equips the relative information contribution (RIC) criterion that indicates when to stop the search procedure. RIC normalized by the sampling numbers n_{sample} until finding n_{new} is defined as

$$I_{\mathrm{RIC}} \triangleq \frac{1}{n_{\mathrm{sample}}} \left(\frac{I_{\mathrm{new}}}{I_{\mathrm{near}}} - 1 \right), \tag{3.18}$$

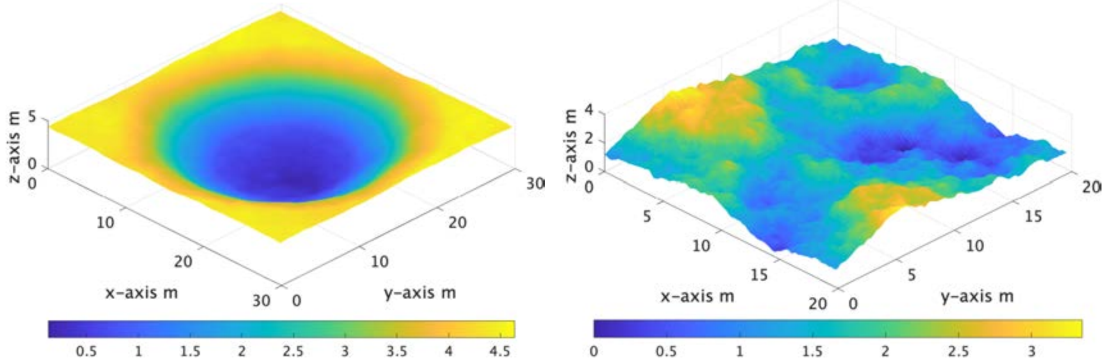
where I_{new} and I_{near} are the measured values from the `Information` function at their corresponding nodes. Equation (3.18) measures how to decrease the relative information gain by expanding the tree, which eventually reaches a small value. A convergence threshold is provided in IIG, but it is vulnerable to changes in reachable areas in the environment. Rather than using an absolute criterion, CC-IIG instead evaluates the relative change of I_{RIC} by calculating the smoothed gradient from the `GradientRIC` function until it reaches zero.

3.5 Simulation Studies

This section demonstrates the proposed active traversability learning framework through simulations in two types of environmental conditions: crater-type and random terrain environments (Fig. 3.3). The crater-type environment in Fig. 3.3 (a) is used to qualitatively verify that the framework learns traversability while avoiding unacceptable immobilized risk. The terrain consists of a benign bottom (0°–5°) to a hostile inner rim (20°–30°) until reaching its outer edge (0°–15°). The local profile has uneven height information generated by the fractal approach in [70] to simulate planetary surface roughness. Conversely, the random terrain environment in Fig. 3.3 (b) provides quantitative algorithm assessments (including the generalization performance) of various terrain geometry and the influence of the predefined parameters. Fractal terrain modeling from [70] is globally applied to generate the randomly distributed terrain inclination. A 0.2 × 0.2 m grid resolution was set as the terrain map parameter.

In the environments, we define loose regolith uniformly covering the terrain surface. The latent function $f_\bullet$ in each direction is given by a logistic equation as follows:

$$f_\bullet(\theta_\bullet) = \frac{100}{1 + \exp\left(-\frac{c_1}{\theta_\bullet^{\max}} \left(\theta_\bullet - c_2 \theta_\bullet^{\max} \right) \right)} + f_{\bullet,0}, \tag{3.19}$$



(a) crater-type terrain (b) random terrain

Fig. 3.3: Virtual terrain environments for the simulation studies. The color indicates the relative height information for each grid.

where $\theta_{\bullet}^{\max}$, $f_{\bullet,0}$, c_1, and c_2 express the slope that induces complete rover stuck, the offset value, and the parameters to tune the function response to $\theta_{\bullet}$, respectively. These values are set as $(\theta_x^{\max}, \theta_y^{\max}) = (30°, 25°)$, $f_{\bullet,0} = 5$, $c_1 = 10$, and $c_2 = 0.7$. When (3.19) receives a slope $\theta_{\bullet}$, the environments return $s_{\bullet}$ with uniformly distributed random noise. Note that (3.19) is assigned as an illustrative example of the latent function, which has nonlinear trends, and the given parameters are also hidden information for the algorithm.

Fixed hyper-parameters were provided in all experiments for the designed covariance function in (3.8) to calculate information quality and update the GPs. While the algorithm basis is identical, adaptive hyper-parameter optimization as shown in [55] is an interesting research direction for facilitating the exploration procedure. We also set the predefined values $\delta_{\mathrm{ME}} = 0.1$ as the termination condition in Algorithm 1 and $s_x^{\mathrm{th}} = s_y^{\mathrm{th}} = 80$ %, expressing maximum allowable slips in (3.13). Lastly, the algorithm assumes safe regions $(\theta_x^{\mathrm{safe}}, \theta_y^{\mathrm{safe}}) = (6°, 5°)$ for the initial exploration procedure. Given parameters, the algorithm implemented in MATLAB runs the exploration tasks in the simulated environments.

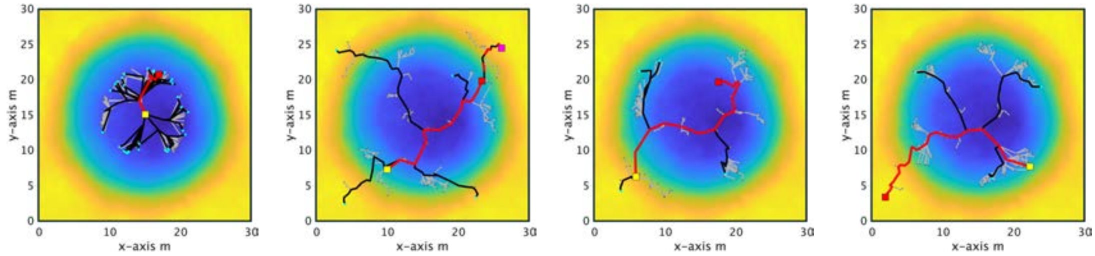
3.5.1 Qualitative Analysis: Crater-type Terrain

In the crater-type terrain, the algorithm explores the 30 $\times$ 30 m environment (Fig. 3.3 (a)) from the bottom position $(x, y) = (15, 15)$ with the given confidence level $\delta_{\mathrm{safe}} = 0.99$. Fig. 3.4 shows how the framework updates the estimated traversability models by generating informative trajectories and taking traverse data while assessing the feasibility of the given trajectories. In Fig. 3.4 (a)–(d), CC-IIG generates the informative tree where grey, black, and red colors indicate complete expansion results, informative trajectories, and the best executed trajectory, respectively. The framework iterates the path planning procedure with CC-IIG 18 times, including initial exploration within the safe region. Fig. 3.4 (a) and (e) show that the GP estimates perform poorly after the initial exploration as CC-IIG only searches the gentle terrain. Subsequently, the algorithm guides the rover beyond the specified safe area, as shown in Fig. 3.4 (b) and (f), by exploiting the probabilistic feasibility assessment driven by the ongoing GP estimates. The framework sequentially refines the GP estimates while triggering replanning when the upcoming state violates constraints (Fig. 3.4 (b)). The estimated traversability model after the final GP update process is shown in Fig. 3.4 (h). The GP root-squared-mean errors (RSMEs) for the hidden models are 1.55 and 1.78, measuring at maximum 75.1 % and 72.9 % slips in each direction. The quantitative results imply that the framework eventually outputs well-learned GPs without violating the given allowable slips. In addition, the GP learning process provides the framework with an ordered nature response to risk, enabling the cautions gathering of traverse data. The organized learning structure allows the CC-IIG to investigate informative but possibly unsafe traverse data, thus avoiding insufficient measurements on hazardous terrain.

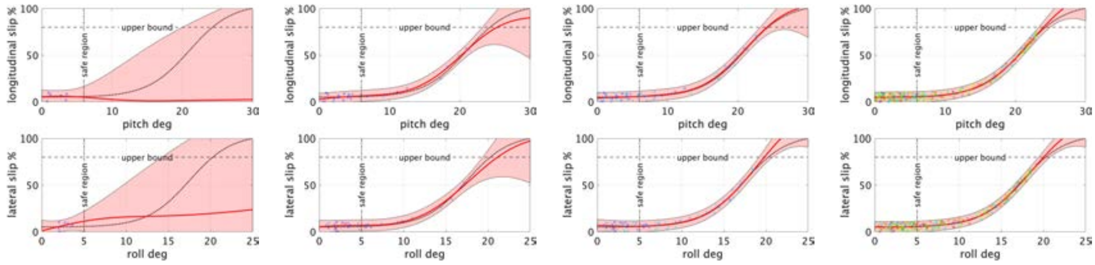
3.5.2 Quantitative Analysis: Random Terrains

For random terrains, the algorithm explores the 20×20 m environment (Fig. 3.3 (b)) starting from the arbitrary benign position with the given confidence level. Here, δ_{safe} was varied to 0.99, 0.9, and 0.6 to run experiments over 50 randomly generated environments. The algorithm was also tested without the `isProbFeasible` function to verify the validity of chance constraint formulation. After running exploration, RMSE and the Kullback-Leibler divergence (KLD) [37] were used to evaluate traversability learning performance. RMSE measures how trained GP models fit the latent traversability model, whereas KLD assesses the similarity of the obtained predictive distribution to the real distribution. Besides, the observed maximum slip and a violation rate are measured to assess relative change in risk awareness with different δ_{safe}. The violation rate represents the percentage of traversed edges where the rover experiences slip above $s_{\bullet}^{\text{th}}$.

Fig. 3.5 illustrates the active traversability learning result in one of the 50 environments. The framework executes 19 CC-IIG iterations with $\delta_{\text{safe}} = 0.6$ that brings more aggressive maneuvers than demonstrated in Section 3.5.1, as measured at most 84.4 %



(a) initial trajectory (b) 3rd trajectory (c) 5th trajectory (d) final (18th) trajecotry

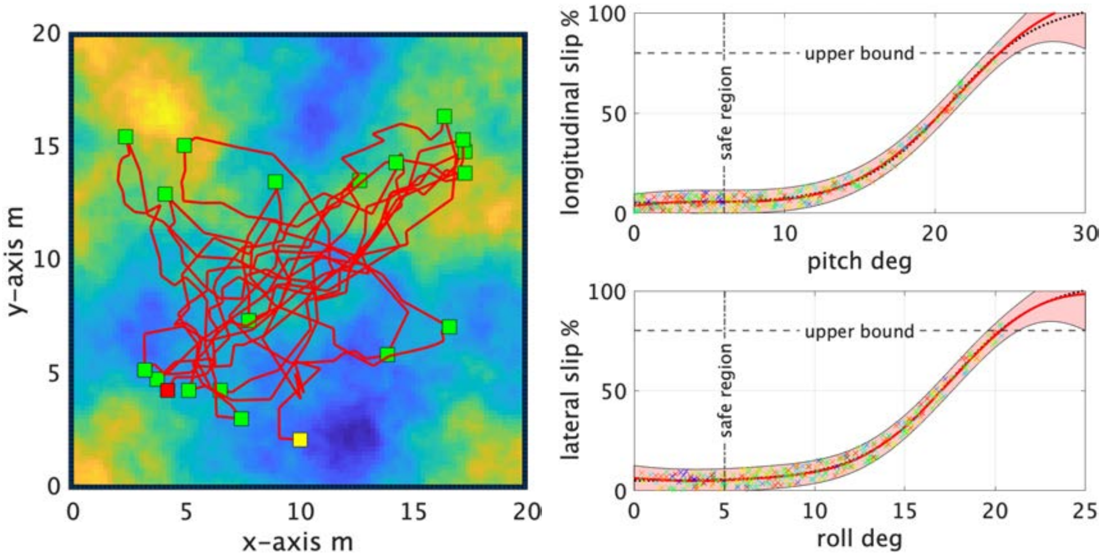


(e) initial model (f) model after 3rd iteration (g) model after 5th iteration (h) final model

Fig. 3.4: Active traversability learning results in the crater-type terrain environment. (a)–(d) denote executed trajectories (red line) with the generated trees by CC-IIG. Yellow and red squares are starting and actual ending states; the magenta square is a planned ending state. (e)–(h) show estimated traversability models with GPs (red line: mean, red-shaded area: 2.5 standard deviations) in longitudinal and lateral directions. The black dotted curve is the latent function, and crosses are gathered *in situ* data colored for expressing iteration numbers on a cool to warm color scale.

and 83.3 % slips in each direction.

Table 3.1 summarizes the traversability learning results for different δ_{safe} and compares them with no stochastic constraints. As shown in the table, the framework can significantly reduce violation rates with uncertainty consideration. Chance constraint formulation is necessary for the framework to gather traverse data while avoiding rover stuck situations. The table also indicates that lower δ_{safe} leads to better convergence of the predictive distribution by gathering greater slips. The lower confidence level allows reaching hazardous states to reduce uncertainty, while a higher value more rigorously avoids immobilized risk. Although the difference in δ_{safe} causes relative performance change in the trained GPs, RMSEs in the table indicate that these GPs achieve robust latent function prediction. By adding the traversability trend with increasing slope gradient into the covariance function in (3.8), the GPs can stably adapt the general correlation of rover-terrain interaction. The quantitative outcomes demonstrate 1)



(a) traversed trajectories (b) final model

Fig. 3.5: Example of active traversability learning results in the random terrain environment. Green squares are ending states for each iteration.

Table 3.1: Traversability learning performance over 50 random terrain environments

	δ_{safe}	RMSE	KLD	Max. slip	Viol. rate
long.	0.99	0.67±0.50	177.2±98.7	73.9±2.2	0.00±0.00
	0.90	0.69±0.68	154.0±146.8	80.9±6.5	0.10±0.16
	0.60	0.53±0.49	92.9±83.8	87.2±7.7	0.54±0.39
	w/o	0.14±0.09	18.6±1.5	100.0±0.0	14.17±2.53
lat.	0.99	1.38±0.83	265.5±182.6	73.8±2.3	0.00±0.00
	0.90	0.81±0.63	115.6±108.9	84.7±8.3	0.18±0.16
	0.60	0.59±0.45	65.2±63.4	89.6±7.8	0.61±0.37
	w/o	0.14±0.12	13.4±1.8	100.0±0.0	22.69±3.08

the effectiveness of stochastic risk assessment for traversability learning, 2) the ability to control the framework behavior with simple parameter adjustment, and 3) robust traversability prediction performance due to the designed covariance function.

3.6 Summary

We have presented an active traversability learning framework to estimate the latent hazardous process via GPs. The proposed two-stage framework first employs the CC-IIG algorithm to search informative trajectories and then takes measurements to update the GPs. Throughout planning and learning, chance constraint formulation exploits current probabilistic estimates to assess immobilized risk within a given confidence level. Simulation studies on rough terrain environments demonstrate that as GP estimates reduce uncertainty, the framework can reach more hazardous states along feasible trajectories in an ordered manner while averting rover entrapment situations. The quantitative analysis further shows the importance of stochastic risk assessment for traversability learning, the ability to control aggressive maneuvers by tuning confidence levels, and robust prediction performance owing to the designed covariance function.

Future work will efficiently guide rovers toward desired destinations while actively learning traversability. Goal-oriented motion planning is essential for implementing the traversability model to achieve safe and efficient autonomous rover operation. Another option is extending the proposed framework to handle mixed terrains by sharing accumulated knowledge with other distinctive materials. The domain transferring approach [46] and vision-based terrain classification [42, 43] could be incorporated to challenge more unstructured environments.

Chapter 4
Probabilistic Traversability Fusion for Risk-aware Path Planning

4.1 Introduction

Reliable rover autonomy is crucial to exploring celestial surfaces, as manual rover operation from Earth involves significant communication latency. Past rovers, such as those for missions on Mars [15], were thus equipped with onboard systems that perceive surrounding environments from stereo vision, assess geometric obstacles, and navigate themselves on collision-free trajectories. Nevertheless, many human interventions were required to assess *traversability* in unexplored environments, resulting in slow rover travel. For instance, the Mars Science Laboratory mission reports an average drive distance of the Curiosity rover being limited to 28.9 meters in one Martian solar day (24 hours 39 min [71]), even though it can travel up to 15.12 m/h [14]. On extraterrestrial terrain, what turned out to be hazardous for rovers other than apparent obstacles were deformable surfaces. As a known case, the Curiosity rover experienced significant slips on rippled sand at the Hidden Valley, forced to change its route for more solid terrain. Such excessive wheel slips degrade driving speed, increase energy consumption, and eventually cause permanent entrapment in loose, granular materials. Hence, reliable traversability assessment on deformable terrains is essential for autonomous rover operation and, in turn, for faster, more extended rover exploration.

Traversability assessment for deformable terrains is challenging due to complex dependencies between physical properties, surface geometry, and rover mobility mechanisms [19]. This problem becomes more complicated when the terrain is *heterogeneous*,

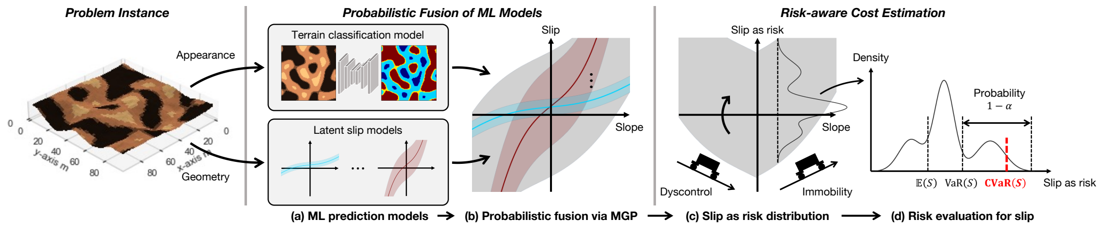


Fig. 4.1: Risk-aware path planning via probabilistic fusion of traversability prediction models and conservative risk evaluation.

i.e., such dependencies even differ from place to place. Thus, instead of theoretical traversability modeling, existing studies attempt to predict traversability via appearance and geometry information using machine learning (ML) algorithms. The appearance information provides cues of surface characteristics, such as material composition, to symbolically categorize heterogeneous terrains via ML classifiers [21, 22, 72, 73]. The geometry information is also given for learning correlations between terrain geometry and slip behavior to predict traversability for unexplored regions [1, 26, 44, 46, 74].

We here argue that for the sake of safe rover navigation, learning-based traversability assessment requires increased attention to potential error inevitably existing in ML predictions. While recent ML models demonstrate significant improvement for robotic perception and control [75–77], they cannot guarantee perfect predictions. This common issue of ML can cause unrecoverable rover immobilization. Moreover, highly accurate traversability prediction is expectedly difficult, particularly for distant planets, due to the presence of unobservable terrain conditions and limited observations for unexplored regions.

To address future *immobilization risk* for safe rover navigation, we devise a path planning algorithm incorporating rational risk assessment against potential error in traversability prediction for heterogeneous terrains. The key idea is the probabilistic fusion of prediction models while taking into account their uncertainties. Suppose we have a terrain classifier predicting heterogeneous terrain classes given appearance information and latent slip (LS) models mapping geometric features to slip behaviors depending on terrain classes via Gaussian processes (GPs) [66] (Fig 4.1a). Our strategy then integrates multiple LS models of all possible terrain classes to form a single, multimodal slip distribution as mixtures of Gaussian processes (MGP; Fig. 4.1b). This retains the prediction uncertainties of each model and enables us to effectively evaluate the immobilization risk via conditional value at risk (CVaR) [78] for conservative path planning (Fig. 4.1c-d).

We extensively evaluate our algorithm in synthetic celestial environments, which pose various difficulties in the traversability prediction. Comparison with different prediction models and risk evaluation metrics validates that our algorithm can plan feasible and safer paths on heterogeneous deformable terrains.

4.2 Related Work

Numerous studies have tackled terrain traversability prediction by employing ML algorithms with visual and geometric information. Classification-based approaches categorically distinguish terrain types, typically traversable or non-traversable regions, from appearance [72] and infrared [73] imagery. Combining vision and proprioceptive (*e.g.*, vibration) sensing has also been studied for increasing robustness against appearance changes [21, 22]. Regression-based approaches assess traversability to a more detailed degree. These approaches model (non-)traversability as the degree of wheel slip and estimate a mapping from terrain inclination to slip via regression algorithms such as GPs [1,26,44,46,74]. To account for multi-modal slip behaviors with regression-based approaches, classify-then-regress-based methods first classify terrains from appearance imagery and then exploit class-dependent regressors for geometry-based slip prediction [1, 44, 46]. Other work exploits a mixture-of-experts (MoE) framework to incorporate terrain class likelihoods for multi-modal slip prediction [26, 74].

These traversability prediction techniques are further utilized in rover navigation studies for planetary exploration. For example, Ono *et al.* [43] use a terrain classifier to detect non-traversable regions for collision-free path planning using the A* algorithm. Similarly, a path planner by Helmick *et al.* [42] exploits a traversability map predicted by an MoE-based regressor. Such direct use of traversability prediction results for rover navigation, however, exposes rovers to the immobilization risk owing to ML-inherent prediction error.

Such prediction error in ML has also been studied under the concept of *uncertainty*. When quantized via probabilistic models, uncertainty provides useful risk assessment tools in ML applications. Particularly in the context of risk-aware path planning for planetary environments, planners based on chance constraint formulations have been proposed by exploiting the variance of GPs as traversability prediction uncertainties [64, 79, 80]. However, these methods rely on single GP models and thus assume homogeneous materials covering given environments or accurate terrain classification results in advance. Moreover, although chance constraints are suitable for capturing risk corresponding to boolean events (*e.g.*, collisions) [78], the immobilization risk incurs a range of costs corresponding to the degree of slip.

Conditional value at risk (CVaR), defined to provide a pessimistic estimate of a random variable by accounting for its uncertainty, is another statistical technique for risk assessment. CVaR satisfies certain axioms required for rational risk assessment for robotics applications under uncertainty [78]. For risk-aware planning, CVaR has been applied to avoid various risks such as moving obstacles in robot navigation [81, 82], uncertain regions in semantic segmentation of road scenes [83], or an abstract risk in a hazardous process [84]. While these studies focus on either collision avoidance [81, 82], classification-based traversability modeling [83], or a single-modal risk behavior [84], our focus is on handling the immobilization risk that requires continuous and multi-modal modeling of rover slip on various kinds of deformable terrains. Risk-aware off-road navigation by Cai *et al.* [85] uses CVaR to estimate pessimistic robot speed in environments with dirt and vegetation. Although their use of CVaR for velocity prediction is similar to ours, their semantic-based scene understanding without considering terrain geometry is unsuitable for capturing the immobilization risk owing to rugged celestial surfaces.

In summary, our work focuses on path planning of rovers in the face of the immobilization risk due to heterogeneous and deformable terrains. Compared to existing planners for planetary rover navigation [42, 43, 64, 79, 80], our method simultaneously considers multi-modal slip behaviors due to heterogeneous terrains and the immobilization risk. The proposed solution (*i.e.*, MGP-based traversability model integrating visual and geometric information as well as CVaR-based continuous (non-boolean) risk assessment) addresses challenges that are only partly handled by existing CVaR-based risk-aware planners in other areas [81–85].

4.3 Preliminaries

This section describes a general path planning formulation, GPs, and CVaR as technical foundations of our algorithm.

4.3.1 Path Planning

The path planning objective is to find a sequence of feasible robot state transitions that navigates a robot to a destination safely and efficiently. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a grid map representing an environment in a 2D space, where $\mathcal{V}$ is a finite set of vertices v enumerating possible robot states and $\mathcal{E}$ is a set of edges representing state transitions from each vertex to its eight neighbors. Each edge is associated with a strictly positive cost calculated by a movement cost function $f_{\text{cost}}(v, v')$. The optimal path planning problem is formulated to find a path $\mathcal{P} = \left\{v_1, v_2, ..., v_{|\mathcal{P}|}\right\}$, from the start $v_1 = v_{\text{start}} \in \mathcal{V}$ to the goal $v_{|\mathcal{P}|} = v_{\text{goal}} \in \mathcal{V}$, with the minimum possible total cost defined as follows:

$$\min_{\mathcal{P}} \sum_{i=1}^{|\mathcal{P}|-1} f_{\text{cost}}\left(v_i, v_{i+1}\right). \tag{4.1}$$

4.3.2 Gaussian Process for Traversability Modeling

Wheeled rovers experience slip when traveling on deformable terrains. We consider the longitudinal slip ratio s to quantitatively describe the rover traversability as follows:

$$s = \begin{cases} \left(u_{\text{ref}} - u\right)/u_{\text{ref}}, & u \leq u_{\text{ref}} : \text{driving}, \\ \left(u_{\text{ref}} - u\right)/u, & u > u_{\text{ref}} : \text{braking}, \end{cases} \tag{4.2}$$

where u and u_{ref} are the actual and reference velocities in the longitudinal direction, respectively [19]. A positive s represents a traverse slower than commanded, with $s = 1$ representing complete rover entrapment in deformable terrains. Conversely, a negative s expresses a traverse faster than commanded.

To model the unknown relationship between terrain geometry and wheel slip s, we exploit GP, a non-parametric regression approach employing statistical inference to learn dependencies between points in a dataset [66]. The use of GPs is justifiable as 1) they can handle the effects of unobservable terrain conditions as uncertainty, and 2) they can express nonlinearity in rover-terrain interactions. A training dataset for GPs may be collected as rovers measure terrain pitch angles ϕ and corresponding longitudinal slips s in the past traverse experience. To account for various slip trends on heterogeneous terrains, suppose terrain classes dominate these slip trends. We then introduce a class-dependent GP that probabilistically models the traversability (*i.e.*, slip s) at an edge $e \in \mathcal{E}$ for a terrain class c as

$$\mathbb{P}_c\left(s|\phi_e\right) := \mathcal{N}\left(\mu_c(\phi_e; \boldsymbol{\phi}, \boldsymbol{s}), \sigma_c^2(\phi_e; \boldsymbol{\phi}, \boldsymbol{s})\right), \tag{4.3}$$

where the predictive mean μ_c and and variance σ_c^2 are parameterized with training samples of input pitch angles $\boldsymbol{\phi}$ and output slip measurements $\boldsymbol{s}$ (see [66] for details).

4.3.3 Conditional Value at Risk

Rovers must measure uncertainty in traversability to assess risk during planning. Unlike collision risk considered in typical planning problems [62], our immobilization risk by deformable terrains gradually increases with slip magnitude. We exploit CVaR to quantify such risk through probabilistic traversability modeling. CVaR measures a conservative expected value accounting for tail events in a given distribution, enabling rational risk assessment without being too optimistic or pessimistic [78]. Let S be a random variable of the longitudinal slip ratio at an arbitrary rover pitch. CVaR at level $\alpha \in [0, 1]$ denotes the expected value of S in the upper $(1-\alpha)$-tail, subject to the conditional distribution of S as follows:

$$\mathrm{CVaR}_\alpha(S) := \mathbb{E}\left[S|S > \mathrm{VaR}_\alpha(S)\right], \tag{4.4}$$

where VaR_α is the value at risk (VaR) expressing $(1-\alpha)$-quantile of S as $\mathrm{VaR}_\alpha(S) := \min\left\{s|\mathbb{P}\left(S > s\right) \leq \alpha\right\}$. Intuitively, CVaR provides pessimistic risk estimates from the given traversability prediction, as illustrated in Fig. 4.1d, factoring in its uncertainty for risk-aware path planning. Given α controls risk assessment behavior from aggressive to conservative throughout the above formulations. If $\alpha = 0$, CVaR returns the expectation. If $\alpha = 1$, the CVaR outcome matches the worst case.

4.4 Risk-aware Path Planning Algorithm

4.4.1 Overview

This section presents the unified algorithm translating ML-based traversability predictions and their uncertainties as risk-aware traveling costs evaluating immobilization risk in planning. Unlike typical path planning problems, such as shortest pathfinding in 2D maps, our formulation for rover traveling requires variable traversability costs to account for slip behaviors on heterogeneous deformable terrains. Although the actual slip trend is unknown, it chiefly depends on the material composition and surface geometry in given environments. We thus associate the environment graph with appearance and geometry information, particularly the colors and 3D positions of terrain surfaces, to be utilized for traversability cost prediction via ML models. Projecting such information into the graph vertices is a durable assumption for actual exploration missions, as the HiRISE camera used for Mars exploration can capture overhead imagery with at most 0.25 m/pixel resolution [86].

The overall method pipeline is illustrated in Fig. 4.1 and summarized as follows. We employ two kinds of pre-trained models: 1) a terrain classifier to pixel-wisely predict terrain classes from appearance imagery and 2) GPs to model class-dependent LS functions with geometry information. These models are fused into a single, multi-modal probabilistic slip distribution via mixtures of GPs (MGP) [87] (Section 4.4.2). We then extract CVaR estimates from this MGP to derive an edge-wise cost for planning (Section 4.4.3). We here essentially translate a probabilistic slip prediction with uncertainty into a deterministic, effectively conservative cost by considering immobilization risk due to the potentially erroneous classifier and GPs. Finally, we perform path planning procedures, such as A*, to find a trajectory that will not induce rover entrapment.

4.4.2 Probabilistic Fusion of ML Models

Here, we elaborate on MGP representing the probabilistic fusion of the appearance-based terrain classifier and geometry-based slip regressors (GPs). The MGP estimates $\mathbb{P}_{e:v \to v'}(s)$, a probability distribution of the slip s in the transition e from a vertex v to its neighbor v'. This distribution is formulated as the sum of multiple class-dependent GPs $\mathbb{P}_e(s|c)$, weighted by classification likelihoods $\mathbb{P}_v(c)$ as

$$\mathbb{P}_{e:v \to v'}(s) = \sum_{c \in C} \mathbb{P}_v(c)\, \mathbb{P}_e(s|c), \tag{4.5}$$

where $\mathbb{P}_e(s|c) = \mathbb{P}_c(s|\phi_e)$ in eq. (4.3). This single MGP distribution in eq. (4.5) can explain various slip trends on heterogeneous terrains and can simultaneously express overall uncertainties from both appearance-based terrain classification and geometry-based slip regression.

For terrain classification, any model taking appearance imagery and predicting a categorical distribution at each pixel can be used. In this work, we leverage the U-Net [88], one of the popular computer vision models for pixel-wise semantic segmentation tasks. Applying this model results in per-pixel class probabilities expressed via the softmax function as $\mathbb{P}_v(c) = \exp(a_c(v)) / \sum_{c' \in C} \exp(a_{c'}(v))$, where $a_c(v)$ is the scalar output of U-Net for a vertex v and a terrain class c. Note that selecting a single GP corresponding to the most likely class $c^\star = \arg\max_c \mathbb{P}_v(c)$ in eq. (4.5), instead of summing multiple GPs, reduces the presented traversability prediction to existing one [1]. However, such a formulation ignores classification-owing uncertainties in the risk assessment described next.

4.4.3 Risk-aware Cost Estimation

To perform path planning using eq. (4.1), we derive its traversability costs f_{cost} from the slip distribution given by MGP. We design the cost function as *risk-aware travel time*, which considers both traversability in terms of risk endangering rovers and traverse efficiency. Travel time is a desirable criterion for the cost function since it encompasses efficiency and safety in terms of maintaining a higher traverse rate with lower slip. The time to traverse an edge $e : v \to v'$ is given as $t = ||\boldsymbol{x}(v) - \boldsymbol{x}(v')||/u(e)$, where $\boldsymbol{x}(v)$ denotes the 3D position at a vertice v and $u(e)$ is the rover velocity when traversing an edge. Here, $u(e)$ is derived with slip s by converting eq. (4.2) as follows:

$$u(e) = \begin{cases} (1-s)\, u_{\text{ref}}, & \phi_e \geq 0 : \text{ascending}, \\ u_{\text{ref}} / \left(1+s\right), & \phi_e < 0 : \text{descending}, \end{cases} \tag{4.6}$$

where ϕ_e is the pitch angle at the edge e. In the ascending case in eq. (4.6), a larger slip $s \in [0, 1]$ can be interpreted as risk since it slows down rover velocity, increases traverse time, and eventually triggers rover immobilization. In the descent case, a negatively larger slip $s \in (-1, 0]$ may lead to faster velocity and shorter travel time. However, this seemingly 'preferable' state should be actually regarded as another risk because it induces rover dyscontrol, such as deviation from the planned trajectory and tip-over situation.

To comprehensively evaluate these two types of risks from slip phenomena, we introduce *slip as risk. Given a random variable of slip $S(\phi)$ parameterized by a pitch angle ϕ, the slip as risk is defined as another random variable as*

$$S_{\text{risk}}(\phi) = \begin{cases} S(\phi), & \phi \geq 0 : \text{ascending}, \\ 2S(0) - S(\phi), & \phi < 0 : \text{descending}. \end{cases} \tag{4.7}$$

Here, the distribution of $S(\phi)$ follows MGP in eq. (4.5). Essentially, the slip as risk in eq. (4.7) evaluates the deviation from the state of traversing on the flat ground (i.e., $\phi = 0$), by vertically flipping $S(\phi)$ in $\phi < 0$ (Fig. 4.1c), assuming that the flat-ground traversing is most stable and least risky. We then extract a pessimistic estimate of $S_{\text{risk}}(\phi_e)$ via CVaR as

$$s^{+}_{\text{risk}}(e) = \text{CVaR}_{\alpha}\left(S_{\text{risk}}(\phi_e)\right). \tag{4.8}$$

By substituting $s = s^{+}_{\text{risk}}$ into eq. (4.6), we obtain a rover velocity accounting for the comprehensive risks, denoted as $u_{\text{risk}}(e)$. The cost function is derived as the risk-aware travel time that evaluates time efficiency and rover traverse stability in both ascent and descent directions, as follows:

$$f_{\text{cost}}(e : v \to v') = \frac{||\boldsymbol{x}(v) - \boldsymbol{x}(v')||}{u_{\text{risk}}(e)}. \tag{4.9}$$

Note that MGP in eq. (4.5) can be alternatively expressed with $\mathbb{P}_{v'}(c)$ instead of $\mathbb{P}_{v}(c)$. We thus compute costs in two ways using both definitions and use their average for planning.

4.5 Simulation Experiments

This section demonstrates the proposed path planning algorithm through extensive simulations and validates its robustness against unreliable traversability prediction.

4.5.1 Datasets

We prepare three synthetic datasets for path planning problems with varying difficulties in traversability prediction. Each dataset has the training, validation, and testing problem instances as 96 × 96 grid maps with a resolution of one grid per meter, providing RGB and height maps as inputs for planners. As hidden information, each instance has pixel-wise assignment of terrain classes $c \in C$ and a set of class-dependent slip functions $f_c(\phi)$. Throughout the dataset creation, fractal terrain modeling [70] is applied to generate randomly distributed rough terrain geometry.

To synthesize diverse but reasonably controlled datasets, we here define *environment groups*, a random subset of terrain classes with occupancy ratios for controlling the terrain class occurrence in each map (*e.g.*, 50% for class #1, 30% for class #4, and 20% for class #6). Training, validation, and testing subsets individually have ten distinct environment groups (*i.e.*, ten different random choices of terrain classes), from each of which we generate 100, 50, and 10 problem instances, respectively. This results in 1000 training, 500 validation, and 100 testing instances. For each problem instance, we first synthesize a random Perlin noise image, cluster the values based on the occupancy ratios specified by the environmental group, and assign a corresponding terrain class (*i.e.*, slip model) to each cluster. For each slip model, a slip measurement process is simulated via additive zero-mean Gaussian noise of a pre-defined variance as $s = f_c(\phi) + \epsilon_c$. Synthesized slip measurements are used to train each class-dependent GP. In-situ measurements are also taken during path executions by substituting calculated ϕ_e into s.

With the above approach in mind, here is the detail of the three datasets we have created.

- **Standard (Std) Dataset:** This dataset adopts a simple setting where pixel color information identifies terrain classes and slip measurements contain slight noise ϵ_c. Each environment group consists of four out of ten terrain classes. Fig. 4.2 (a) and (b) show an example terrain appearance image and a corresponding terrain class map coded by distinctive colors, respectively. Fig. 4.2 (c) shows learned slip models via GPs.

- **Erroneous Slip (ES) Dataset:** A more complicated setting is considered in this dataset by imposing greater noise ϵ_c in slip measurements as the gradient magnitude of $f_c(\phi)$ increases. Hence, learned GPs result in erroneous slip models shown in Fig. 4.2 (f). As described in [1], such erroneous slip modeling can happen due to multiple factors, including insufficient slip measurements and unobservable terrain conditions.

- **Ambiguous Appearance (AA) Dataset:** We replicate another complicated setting where multiple terrain classes appear with the same appearance, assuming situations, *e.g.*, when distinctive terrain characteristics do not provide visually identifiable cues or one terrain material is covered by another. This dataset comprises four pairs of terrain classes, each with high-variance GPs (as in the ES dataset) but sharing the same color cues, resulting in eight classes with less appearance diversity. Each environment group is then defined by the random ordering of those classes such that four different colors always appear, as shown in Fig. 4.2 (d) and (e).

4.5.2 Implementation Details

As the path search algorithm, we use the A* search, which returns a resolution-optimal solution as long as it exists in a given discrete domain. As the terrain classifier, we adopt the U-Net model [88] with the ResNet-18 encoder [89] to provide classification

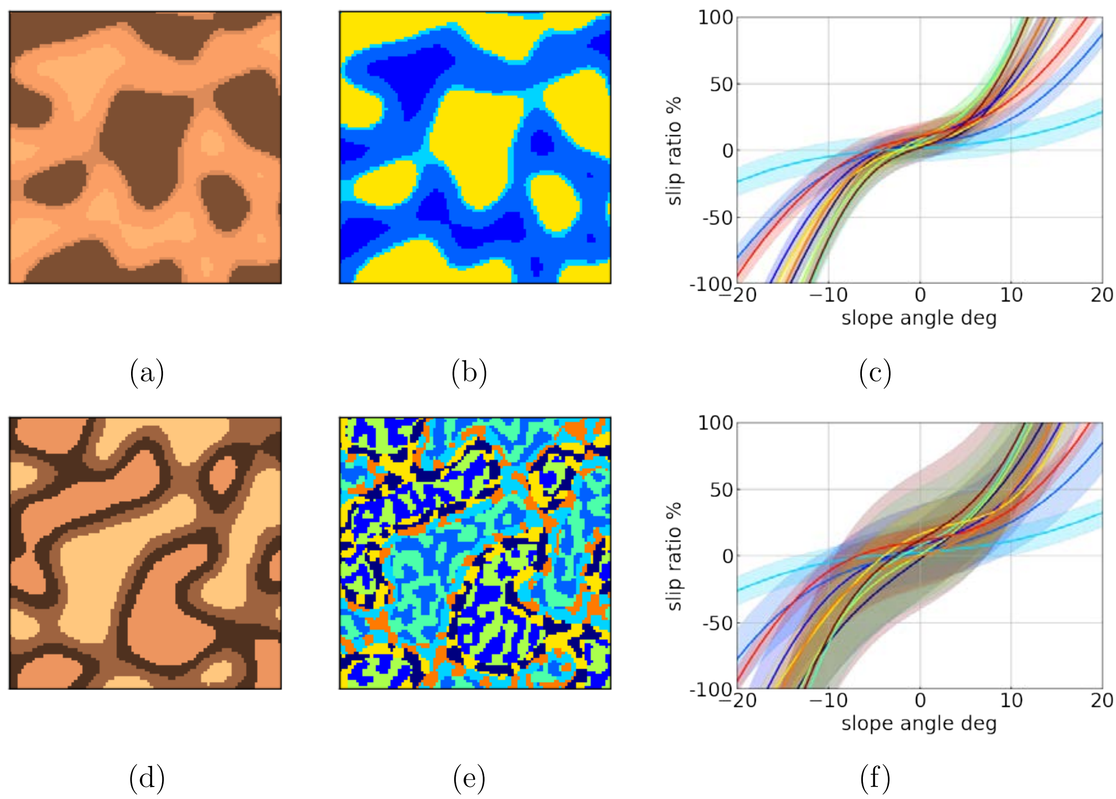
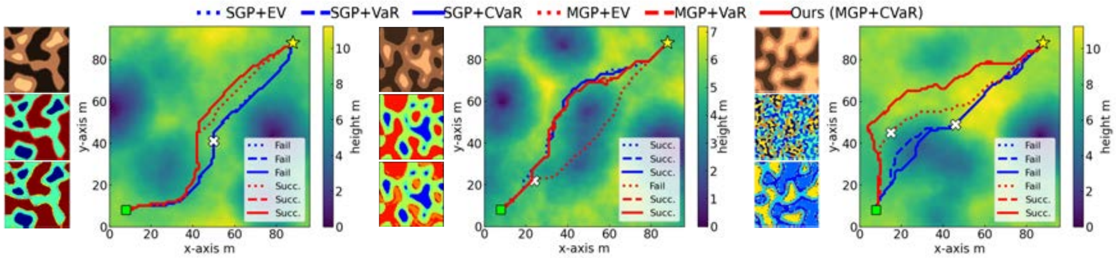


Fig. 4.2: Visualized examples of the (a)-(c) Std dataset and (d)-(f) AA dataset. Each dataset consists of color maps, corresponding terrain class maps, and GPs expressing learned latent slip functions (solid line: mean, shaded area: 95% confidence region). Color in terrain class maps and GPs denote distinct terrain classes.

likelihoods. This model was trained for each dataset separately using its training subset for 50 epochs using the Adam optimizer [90] with a learning rate of 0.001 and a batch size of 8. The model weight producing the lowest loss on the validation subset was used for evaluation on the test subset. Computing VaR and CVaR for random variables associated with the MGP distribution in eq. (4.5) was carried out approximately through a Monte Carlo simulation.

4.5.3 Experimental Setups

In every problem instance, planners search the path from the start position (8 m, 8 m) to the goal (88 m, 88 m). After finding a solution, a rover follows the given path at $u_{\mathrm{ref}} = 0.1$ m/sec while receiving an edge-wise noisy slip s_e. We regard the solution as a success path if we never observe $s_e \geq 1$ (complete rover immobilization) and $s_e \leq -1$ (uncontrollable situations such as tip-over) on the way. The following



(a) Std Dataset (b) ES Dataset (c) AA Dataset

Fig. 4.3: Path planning results over different datasets. In the environments represented as heatmaps, green squares, and yellow stars are start and goal positions. White crosses denote positions where rovers fail to transit to the next states due to experiencing $s = 1$. Input colors, ground truth terrain classes, and terrain classification results are shown from top to bottom on the left side of each heatmap.

four metrics quantify algorithm performances.

- **Solved rate (Sol.)** [%] expresses the ratio of problem instances for which a planner finds solutions.
- **Success rate (Succ.)** [%] denotes the ratio of problem instances whose found solutions are successful.
- **Total time (T_{total}) [min]** provides path efficiency by calculating the total traversing time along the path when traversing is successful.
- **Maximum slip ($s_{\max}$)** [%] evaluates path safety against rover entrapment by calculating $100 \times \max(\boldsymbol{s}_e)$ after path execution, where $\boldsymbol{s}_e$ denotes the experienced slips. A lower $s_{\max}$ indicates a safer traverse on terrains.

To see the effectiveness of our proposed approach, we also evaluated different traversability prediction models and risk evaluation metrics. As for the prediction, a single Gaussian process (SGP) approach is given by selecting a GP based on classified terrain types [1]. In addition to VaR and CVaR, the expected value (EV) quantifies the given slip distribution for cost calculation. With combinations of the above metrics, planning procedures run on the datasets for comparison.

4.5.4 Results and Discussions

Quantitative Comparison

Table 4.1 summarizes quantitative path planning performances on the three datasets, with $\alpha = 0.99$ in VaR and CVaR. Throughout the datasets, our proposed approach

Table 4.1: Quantitative path planning results over different datasets

Standard (Std) Dataset					
		Sol.	Succ.	T_{total}	$s_{\max}$
SGP [1]	+EV	100	71	22.6 ± 3.7	58.3 ± 47.0
	+VaR	100	74	22.6 ± 3.9	55.6 ± 45.4
	+CVaR	100	75	22.6 ± 3.9	55.3 ± 45.8
MGP	+EV	100	75	**22.5** ± 3.5	56.1 ± 44.4
	+VaR	100	93	22.9 ± 4.4	47.5 ± 30.7
Ours (MGP+CVaR)		100	**96**	23.2 ± 5.2	**45.9** ± 24.2
Erroneous Slip (ES) Dataset					
		Sol.	Succ.	T_{total}	$s_{\max}$
SGP [1]	+EV	100	60	**23.0** ± 3.5	70.3 ± 40.7
	+VaR	98	65	24.3 ± 5.1	68.8 ± 44.7
	+CVaR	93	62	25.0 ± 5.7	67.9 ± 44.8
MGP	+EV	100	71	23.1 ± 4.4	68.2 ± 38.2
	+VaR	98	**77**	25.2 ± 12.7	63.1 ± 38.8
Ours (MGP+CVaR)		90	**77**	27.2 ± 37.4	**60.5** ± 37.1
Ambiguous Appearance (AA) Dataset					
		Sol.	Succ.	T_{total}	$s_{\max}$
SGP [1]	+EV	100	11	**22.1** ± 1.7	92.9 ± 27.4
	+VaR	100	7	24.8 ± 4.0	95.8 ± 19.3
	+CVaR	100	2	26.7 ± 1.8	96.3 ± 19.3
MGP	+EV	100	38	24.0 ± 5.6	81.4 ± 32.9
	+VaR	98	**95**	25.7 ± 5.3	64.5 ± 22.3
Ours (MGP+CVaR)		96	**95**	26.4 ± 6.3	**63.7** ± 21.6

owns the highest performance in terms of safety to avert immobilized situations, as shown by its success rates and maximum slips during path execution. Having lower slip also contributes to retaining efficiency in time by maintaining rover velocity, as shown by the total time outcomes. These results are preferable for planetary exploration, where safety is of utmost importance, as rover entrapment leads to mission failure while overly conservative rover navigation delays mission schedules. Erroneous slip modeling, despite higher appearance diversity experienced in the ES Dataset, poses a relative performance decrease in planning outcomes compared to the other datasets. We found that the planner with VaR- and CVaR-based risk evaluations sometimes does not provide solutions. This can happen when the goal position is in hazardous areas. Dismissing such risk to prioritize finding solutions worsens the success rate, as

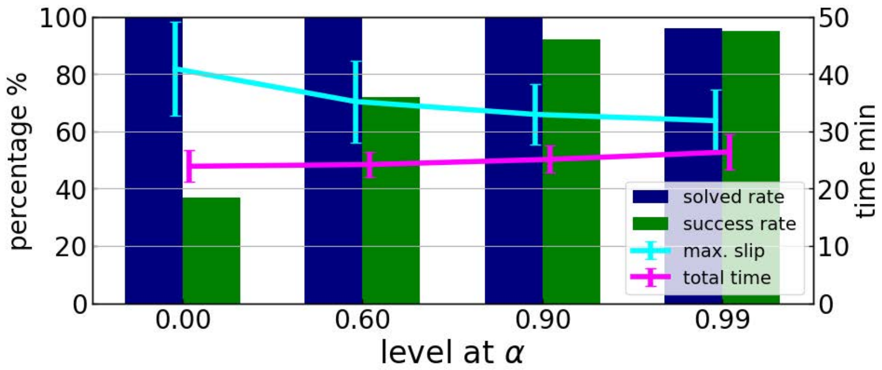


Fig. 4.4: Parameter study results over AA Dataset. Line graphs use mean and standard deviation format.

indicated by the EV-based planners. The MGP formulation for traversability prediction greatly contributes to the success of rover navigation. The most effective cases are those provided by the AA Dataset, where terrain appearance cannot assure its terrain class. Without MGP, traversability prediction only refers to representative slip models according to identified terrain properties, resulting in failed traverses due to huge prediction errors from wrong slip models. Our proposed algorithm solves this issue by integrating multiple GPs with terrain likelihoods, thus enabling careful risk evaluation accounting for less-likely tail events that can lead to immobilized situations. Total computation time for planning, including cost estimation with MGP, is typically 10 to 15 minutes on a standard computer in Python implementation.

Qualitative Comparison

Fig. 4.3 illustrates how the planners with different risk metrics plan paths for a problem instance from each Dataset. MGP-based approaches tend to find trajectories along geometrically benign regions compared to SGP-based ones, but the MGP+EV-based planner often seeks a too optimistic path leading to rover immobilization, as shown in Fig. 4.3 (b). This is because the MGP generates a slip distribution covering a broader range of slips compared to a single GP. VaR and CVaR are thereby necessary for rational risk evaluation to see less likely occurring slips in the MGP distribution, enabling successful rover traverses in challenging environments.

Parameter Study

Finally, a parameter study testing $\alpha \in \{0.00, 0.60, 0.90, 0.99\}$ for the AA Dataset is conducted to see its effects against the performance of the proposed algorithm. Fig. 4.4 indicates that α governs the trade-off between safety versus efficiency: a higher α leads to an increased success rate by navigating rovers on a less-slippery path, while a lower α allows a more aggressive maneuver to shorten the time to traverse. The controllability of planning behavior via simple parameter tuning is a beneficial aspect of our method, considering that priorities of safety and efficiency will change depending on the mission timeline; we may want to prioritize safety during the early exploration phase when most terrain properties are unknown.

Limitations and Possible Extensions

Although our proposed algorithm can significantly improve safely traversing celestial analog environments compared to existing methods, our approach still leaves room for improvement in terms of safety. As shown in Table 4.1, the performance on the ES Dataset demonstrates limitations in our current implementation. Given the use of CVaR, the confidence level and success rate should theoretically align. While we set $\alpha = 0.99$ in Table 4.1, suggesting an expected performance of 99%, the ES Dataset only achieved a 77% success rate. We hypothesize that this performance degradation stems from the limited training data for the terrain classifier, which results in inaccurate terrain class likelihood estimates.

While increasing training data can address this limitation in simulation, real-world applications face inherent challenges. In practical environments, obtaining sufficient annotations for training the terrain classifier is often infeasible, which disturbs well-calibrated uncertainty estimates. This limitation could be addressed through domain adaptation of well-trained models using in-situ measurements to transfer knowledge to new environments. Alternatively, developing annotation-free, self-supervised learning approaches would enable direct learning of traversability from robot experience.

4.6 Summary

We presented a new path planning algorithm that can explicitly consider the uncertainty in ML-based traversability prediction for reliable rover operations on heterogeneous deformable terrains. Through the extensive simulation experiments, we confirmed that the proposed algorithm consistently outperformed existing methods in terms of planning success rates and the maximum degrees of wheel slip experienced during path execution. Future work will investigate the feasibility of the proposed method using actual data such as those for Mars [91]. Another possible future direction is to extend the algorithm to refine the prediction model through online learning [80].

Chapter 5

Robust and Adaptive Uncertainty Handling for Scene Understanding

5.1 Introduction

Autonomy is crucial for rapid and extensive robotic planetary exploration due to significant communication delays between Earth and space. In NASA's Mars 2020 mission, the Perseverance rover's autonomous navigation system (AutoNav [15]) averaged a traverse distance of 144.4 meters per Martian solar day (24 h 39 min [71]). Although this performance surpasses that of prior rovers [92], AutoNav has room for improvement in challenging environments, as it primarily achieved rapid traverse on benign terrain. While AutoNav evaluates *traversability* by detecting geometric obstacles on Martian terrain [15], deformable surfaces covered with fine-grained regolith are more hazardous for rovers than apparent obstacles. These deformable surfaces induce excessive wheel slip, degrade driving speed, increase energy consumption, and eventually cause permanent entrapment in loose regolith. In one case, NASA's Opportunity rover spent more than six weeks trapped in the rippled sand of the Purgatory Dune, resulting in a prolonged mission timeline [93]. Hence, reliable traversability assessment of wheel slip is essential to ensure the safety of rovers traversing unexplored regions.

Traversability assessment of deformable terrain is challenging because of the complex mechanical interactions arising from rover mobility mechanisms and terrain characteristics, such as mechanical properties and surface geometry [19]. Machine learning (ML) can be used to derive latent relationships between environmental features and slip behaviors from training samples. However, despite advances in robot perception

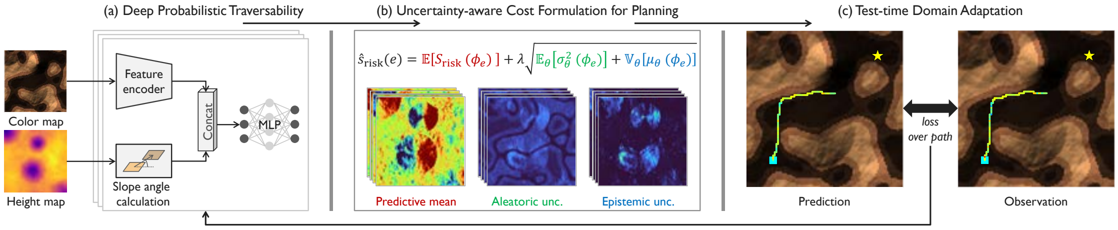


Fig. 5.1: Overview of the proposed unified framework for planetary rover navigation, consisting of (a) deep probabilistic traversability, (b) uncertainty-aware path planning, and (c) test-time domain adaptation with in-situ slip measurements, forming an iterative loop to address uncertain traversability prediction.

and planning [75, 77], no ML model can guarantee perfect predictions owing to inherent *predictive uncertainty*. This uncertainty leads to erroneous predictions, potentially causing the unrecoverable immobilization of rovers on distant planets. Thus, addressing uncertainty in traversability prediction is essential for successful rover navigation.

In ML-based traversability prediction, uncertainty can be managed via three principal approaches. *Quantification* of uncertainty, usually through probabilistic models, provides the degree of uncertainty in the prediction. *Exploitation* of measured uncertainty in planning and decision-making processes helps mitigate the risks of potential prediction errors. *Adaptation* of pre-trained ML models during navigation is also effective in reducing uncertainty due to limited prior observations. Regarding distant planets, ML-based navigation struggles with limited prior traverse experience and uncontrollable environmental factors, such as lighting conditions, which can cause *domain shifts* between training and test environmental conditions.

In this study, we integrate these three uncertainty handling approaches, namely uncertainty quantification and exploitation and model adaptation, into a unified learning and planning framework for safe rover navigation. The key concept in uniting these approaches is *deep probabilistic traversability*, an end-to-end probabilistic ML model using deep neural networks (DNNs) for traversability prediction (Fig. 5.1a). This model predicts the rover's wheel slip directly from visual and geometric observations of the target environment while quantifying the predictive uncertainty via probability distributions. Subsequently, this uncertainty can be exploited to reduce the risks associated with imperfect predictions in path planning (Fig. 5.1b) and is further minimized by adapting the learned distributions to in-situ slip measurements during path execution (Fig. 5.1c).

We extensively evaluate our method in synthetic celestial environments, focusing on the challenges posed by out-of-domain (OOD) geometry and appearance observations. Comparisons with existing methods demonstrate that our method, with uncertainty-aware planning and domain adaptation, achieves safer rover navigation under novel environmental conditions.

5.2 Related Work

ML-based traversability prediction is a key component of off-road autonomy, where *visual* and *geometric* cues are essential. Visual cues capture the diverse geological semantic features that influence robot behavior. Thus, classification methods use the semantic segmentation of appearance imagery to categorize symbolic terrain types and identify traversable areas [72, 94]. This appearance-based classification is often combined with geometry-based regression methods to provide further variation in traversability within each class. Classify-then-regress methods model the traversability for each terrain class by correlating the degree of wheel slip with the terrain inclination [1, 44, 46]. Mixture-of-experts (MoE) methods further account for terrain class likelihoods for slip regressors [2, 26]. Although these combined approaches using visual and geometric cues are effective, they require separate training processes for the classifier and regressor models, which involve human supervision.

Using the measured traverse data for supervision, learning traversability from experience has become a viable alternative. This approach provides a direct interpretation of robot–terrain interactions and eliminates the need for labor-intensive manual annotation. Previous studies often used past driving trajectories to auto-generate ground-truth traversability, which assumes scene familiarity with robots [95–97]. Proprioceptive signals, such as force-torque [98] and linear acceleration [99], define traversability in continuous-valued regression models. The absence of manual annotation also facilitates model adaptation through incremental learning with in-situ measurements, improving the robustness to novel situations [100, 101]. Our method builds on this learning-from-experience paradigm with an end-to-end DNN backbone, while exploiting both visual and geometric cues.

Off-road navigation algorithms leverage traversability-prediction techniques to ensure robot safety in unstructured environments. In planetary exploration, rover navigation systems adopt classification models to identify traversable regions [43] and regression models to derive continuous traversability costs from wheel slip [42]. In terrestrial settings, some studies [102–104] employed traversability models within the model predictive path integral (MPPI) control framework for off-road navigation. However, these methods do not exploit the uncertainty in traversability prediction for safer navigation, as they often rely on deterministic models [102–104]. By contrast, we introduce uncertainty quantification in the prediction models to estimate the reliability of predictions and reduce the risk of rover immobility.

Uncertainty quantification has been extensively studied in ML, including the Bayesian interpretation [66,105], Monte Carlo dropout [106], and model ensembles [107] (see [108] for details). In robotic decision-making, uncertainty quantification offers valuable tools for risk assessment and is used as a constraint or cost. For instance, chance-constraint formulations have been explored for risk-aware off-road navigation [64,79,80]; however, these constraint-based methods are better suited for Boolean events rather than a range of risks, such as slip. Conversely, cost-based methods estimate conservative traversability by incorporating uncertainty into costs [2,84,109–112]. This approach involves designing custom-uncertainty-aware traveling costs [109,110] or applying statistical risk assessments [85,113,114], such as conditional value-at-risk (CVaR) [78]. Other work explores reinforcement learning approaches that can implicitly exploit uncertainty to learn conservative navigation policies [112]. We employ cost-based uncertainty integration to express a range of risks while allowing for hard-obstacle constraints in regions where rover immobilization is likely to occur.

Alternative to uncertainty avoidance, domain adaptation reduces uncertainties and improves predictions by using new observations in test environments. For planetary exploration, Hedrick *et al.* [45] proposed an adaptive navigation system that updates a velocity map online using a Markov random field model. For terrestrial navigation, adaptation using traverse experiences as self-supervision has been studied [100,101]. However, their deterministic predictions, which lack uncertainty modeling, may pose serious risks to planetary robots. Kim *et al.* [115] presented an active domain adaptation approach for probabilistic traversability but it assumes human monitoring during the adaptation phase before deployment. Our deep probabilistic traversability model adapts *on the fly* to in-situ slip measurements during rover path execution while predicting the occurrence of uncertain and risky regions.

Overall, we focus on planetary rover navigation under the risk of immobility due to uncertain slip on deformable terrain. Our findings quantify, exploit, and reduce the uncertainty in traversability prediction within a unified framework, addressing challenges that have only been partially addressed by existing methods [2,85,110,113–115].

5.3 Preliminaries

This section outlines a general formulation for path planning and traversability in terms of wheel slip and presents a target map representation for predicting traversability.

5.3.1 Path Planning

The path planning objective is to determine a series of feasible state transitions that allow a robot to navigate toward its destination safely and efficiently. Consider a grid map $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ representing an environment in 2D space, where $\mathcal{V}$ is a finite set of nodes v enumerating possible robot states, and $\mathcal{E}$ is a set of edges e representing state transitions from each node to its neighbors. We follow a widely used setting, where $\mathcal{G}$ denotes a four-neighbor grid environment with evenly divided grids. Every edge has a strictly positive cost calculated by a travel cost function $f_{\text{cost}}(v, v')$. The optimal path planning problem is formulated to find a path $\mathcal{P} = \left\{v_1, v_2, ..., v_{|\mathcal{P}|}\right\}$, from the start $v_1 = v_{\text{start}} \in \mathcal{V}$ to the goal $v_{|\mathcal{P}|} = v_{\text{goal}} \in \mathcal{V}$, with the minimum possible total cost defined as follows:

$$\min_{\mathcal{P}} \sum_{i=1}^{|\mathcal{P}|-1} f_{\text{cost}}\left(v_i, v_{i+1}\right). \tag{5.1}$$

5.3.2 Traversability as Wheel Slip

Wheeled rovers experience slipping when traversing a deformable terrain. We consider the longitudinal slip ratio s to quantify rover traversability as follows:

$$s = \begin{cases} \left(u_{\text{ref}} - u\right)/u_{\text{ref}}, & u \leq u_{\text{ref}} : \text{driving state}, \\ \left(u_{\text{ref}} - u\right)/u, & u > u_{\text{ref}} : \text{braking state}, \end{cases} \tag{5.2}$$

where u and u_{ref} denote the actual and reference velocities in the longitudinal direction, respectively. A positive slip $s \in [0, 1]$ indicates a traverse slower than the one commanded, with higher values increasing travel time and, ultimately, causing rover immobilization on deformable terrain. Conversely, a large negative slip $s \in (-1, 0]$ indicates a traverse faster than the one commanded, which can lead to rover dyscontrol when $s \approx -1$. Although a higher velocity may seem preferable for reducing travel time, it poses operational risks, including deviations from the planned path and potential tip-overs.

5.3.3 Map Representation

Unlike typical path-planning problems, such as shortest pathfinding in 2D maps, our formulation for rover traveling requires variable traversability costs to account for slip behavior on deformable terrain. While the actual slip trend is unknown, it primarily depends on terrain geological characteristics, such as mechanical properties and surface geometry in the given environments. Thus, we associate the environment graph with the appearance and geometry of the terrain, particularly terrain surface colors and 3D positions, to predict traversability using ML models. Projecting this information onto graph vertices is feasible for exploration missions, as the HiRISE camera used in Mars exploration captures overhead imagery with up to 0.5 m resolution [86].

5.4 Uncertainty-aware Navigation Algorithm

This section presents a unified framework for rover navigation that quantifies, exploits, and reduces the uncertainties to enhance the safety of the navigation process. The overall flow of the method is illustrated in Fig. 5.1 and summarized as follows: Given the appearance and geometry information, we first use the deep probabilistic traversability (DPT) to predict slips and their uncertainties as slip distributions (Section 5.4.1, Fig. 5.1a). Subsequently, we convert these distributions into traversability costs, accounting for potential prediction errors (Section 5.4.2, Fig. 5.1b). These costs enable path-planning searches using (5.1) to determine a path that minimizes the risk of rover immobilization. Finally, we extend this process by incorporating the test-time adaptation of DPT and replanning using in-situ slip observations (Section 5.4.3, Fig. 5.1c).

5.4.1 Deep Probabilistic Traversability Model

We model the DPT using an ensemble of M probabilistic DNNs [107]. We assume an identical architecture for these DNNs with different parameters θ_m for each network, denoted as f_{θ_m}. Each network takes a bird's-eye view environment map of $H \times W$ nodes as the input (Fig. 5.1a, left). The map represents node-wise RGB colors $I_v \in \mathbb{R}^3$ as appearance cues and edge-wise terrain pitch angles $\phi_e \in \mathbb{R}$ associated with the rover's four possible movements at each node as geometric cues. Each network predicts a latent slip distribution for each edge e by predicting the mean and variance of the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ as $[\mu_{e,\theta_m}, \sigma^2_{e,\theta_m}] = f_{\theta_m}(I, \phi_e)$. The DPT then models the slip on each edge e as a random variable S_e following a mixture of Gaussian distributions as

$$S_e \triangleq S_e(\phi_e) \sim \frac{1}{M} \sum_{m=1}^{M} \mathcal{N}(s|\mu_{e,\theta_m}, \sigma^2_{e,\theta_m}). \tag{5.3}$$

The networks f_{θ_m} can accept an arbitrary pitch angle ϕ to emulate slip distributions $S_e(\phi)$ on edge e for hypothetical angles ϕ different from the actual angle ϕ_e. This design is useful for subsequent cost derivations. Thus, f_{θ_m} is implemented in two steps. First, a convolutional neural network (CNN) extracts a feature map from the color map I as $F = \mathrm{CNN}_{\theta_m}(I)$. Subsequently, for each edge $e : v \to v'$, a multi-layer perceptron (MLP) predicts a Gaussian from the feature pair $(F_v, F_{v'})$ and the pitch angle ϕ_e as $[\mu_{e,\theta_m}, \sigma^2_{e,\theta_m}] = \mathrm{MLP}_{\theta_m}(F_v, F_{v'}, \phi_e)$,

The networks f_{θ_m} are initialized differently and trained individually to minimize the negative log-likelihood loss.

$$L_{\theta_m} = -\frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \log \left(\mathcal{N}(s^*_e|\mu_{e,\theta_m}, \sigma^2_{e,\theta_m}) \right), \tag{5.4}$$

where s^*_e is the ground-truth slip from training maps.

Note that the Gaussian mixture distribution in (5.3) has the following forms for the mean and variance:

$$\mathbb{E}(S_e) = \mathbb{E}_\theta[\mu_{e,\theta}], \tag{5.5a}$$

$$\mathbb{V}(S_e) = \mathbb{E}_\theta[\sigma^2_{e,\theta}] + \mathbb{V}_\theta[\mu_{e,\theta}], \tag{5.5b}$$

where $\mathbb{E}_\theta[X_\theta] = \frac{1}{M} \sum_{m=1}^{M} X_{\theta_m}$ denotes the expectation, and $\mathbb{V}_\theta[X_\theta] = \frac{1}{M} \sum_{m=1}^{M} (X_{\theta_m} - \mathbb{E}_\theta[X_\theta])^2$ denotes the variance between the ensemble.

We use the ensemble mean $\mathbb{E}(S_e)$ for slip prediction and the total variance $\mathbb{V}(S_e)$ to quantify uncertainty. In (5.5b), $\mathbb{V}(S_e)$ is decomposed into two components, expressing different types of uncertainties. The first component, the average of the variance estimates, is known as aleatoric uncertainty and reflects the natural variability from the inherent stochasticity of the observations. The second component, the variance of the mean estimates, is known as the epistemic uncertainty and reflects the gaps between the training and test data.

In the next section, we derive uncertainty-aware traveling costs using these predictions and omit the subscript e from S_e, μ_e and σ^2_e by focusing on the same edge e.

5.4.2 Uncertainty-aware Cost Formulation

We formulate the cost function f_{cost} in (5.1) using the DPT predictions provided in the previous section. The edge cost f_{cost} is based on the predicted travel time, which encompasses both efficiency and safety since faster traverses lead to shorter travel times with lower slips.

The travel time for each edge $e : v \to v'$ is given by $t_e = ||\boldsymbol{x}(v) - \boldsymbol{x}(v')||/u(e)$, where $\boldsymbol{x}(v)$ denotes the 3D position at node v, and $u(e)$ denotes the rover velocity during the edge transition. The velocity $u(e)$ can be calculated from the slip s using (5.2) as follows:

$$u(e) = \begin{cases} (1-s)\, u_{\text{ref}}, & s \geq 0 : \text{driving state}, \\ u_{\text{ref}} / \,(1+s)\,, & s < 0 : \text{braking state}. \end{cases} \tag{5.6}$$

Although treating the predicted travel times directly as costs can reduce the risk of rover immobilization (because $s = 1$ results in an infinite travel time), it favors faster rover traverses with larger negative slips, which can endanger the rover. To reduce the risks associated with excessively fast traverse, we adopt the concept of *slip as a risk* introduced in [2]. This concept views any slip state other than that on flat ground ($\phi = 0$) as a potential risk. Thus, it evaluates the deviations from this stable state as positive values by inverting the sign of the slip $S(\phi)$ for $\phi < 0$ as follows:

$$S_{\text{risk}}(\phi) = \begin{cases} S(\phi), & \phi \geq 0 : \text{ascent}, \\ 2\mathbb{E}[S(0)] - S(\phi), & \phi < 0 : \text{descent}, \end{cases} \tag{5.7}$$

where $S(\phi)$ denotes a random slip variable from (5.3).

Using the slip as a risk, we formulate an uncertainty-aware slip estimate $\hat{s}_{\text{risk}}$ as follows:

$$\hat{s}_{\text{risk}}(e) = \mathbb{E}[S_{\text{risk}}(\phi_e)] + \lambda\sqrt{\mathrm{Var}[S_{\text{risk}}(\phi_e)]}, \tag{5.8a}$$

$$= \mathbb{E}[S_{\text{risk}}(\phi_e)] + \lambda\sqrt{\mathbb{E}_\theta[\sigma_\theta^2] + \mathbb{V}_\theta[\mu_\theta]}. \tag{5.8b}$$

This estimate computes the expectation of S_{risk} using the mixture distribution in (5.3) and further pessimistically adjusts this risk estimate by adding the standard deviation of S_{risk} scaled by the hyperparameter $\lambda \in \mathbb{R}^+$.

After converting $\hat{s}_{\text{risk}}(e)$ to a risk-aware velocity estimate $\hat{u}_{\text{risk}}(e)$ via the slip-to-velocity conversion in (5.6), we finally derive our cost as a risk-aware travel time as follows:

$$f_{\text{cost}}(e : v \to v') = \frac{||\boldsymbol{x}(v) - \boldsymbol{x}(v')||}{\hat{u}_{\text{risk}}(e)}. \tag{5.9}$$

This cost function evaluates both the traverse efficiency and stability in ascent and descent, while considering the uncertainties in slip predictions.

5.4.3 Test-time Domain Adaptation

We incorporate test-time domain adaptation into our framework using actual slip experiences observed in-situ during path execution, as illustrated in Fig. 5.1c. Specifically, we perform adaptation after each rover movement and re-plan using the updated DPT predictions. This test-time adaptation is enabled by our end-to-end DPT model f_{θ_m}, which is trained solely from slip experience using the loss function in (5.4), allowing for the backpropagation of the loss computed along the running trajectory.

However, on-the-fly test time adaptation is challenging due to the limited number of available samples. Without sufficient samples, adapting pre-trained models may unintentionally eliminate valuable knowledge acquired during pre-training.

Inspired by previous work on few-shot test-time adaptation [116], we propose adapting pre-trained models f_{θ_m} by adding and tuning a fraction of new parameters while keeping all pre-trained parameters θ_m fixed. Particularly, after every linear and convolutional layer in f_{θ_m}, denoted by $g(\boldsymbol{x}) = W\boldsymbol{x} + \boldsymbol{b}$, we insert a trainable adaptation layer as

$$\bar{g}(\boldsymbol{x}) = \boldsymbol{\gamma} \odot g(\boldsymbol{x}) + \boldsymbol{\beta}, \tag{5.10}$$

where $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are vectors of new trainable parameters, and $\odot$ is an element-wise multiplication. We initialize $\boldsymbol{\gamma}$ to $\mathbb{1}$ and $\boldsymbol{\beta}$ to $\mathbb{0}$, such that this layer initially preserves the original model behavior, that is, $\bar{g}(\boldsymbol{x}) = g(\boldsymbol{x})$.

Compared with directly updating W and $\boldsymbol{b}$ in each layer g, this adaptation model better preserves the original semantics of each feature channel in $\boldsymbol{y} = g(\boldsymbol{x})$ by avoiding the mixing of $\boldsymbol{y}$ across the channels. Additionally, having significantly fewer parameters in $\boldsymbol{\gamma}$ than in W helps prevent overfitting of the models to a limited number of samples.

5.5 Simulation Experiments

In this section, we validate our rover navigation algorithm through extensive simulations to demonstrate its effectiveness under various uncertainties in traversability

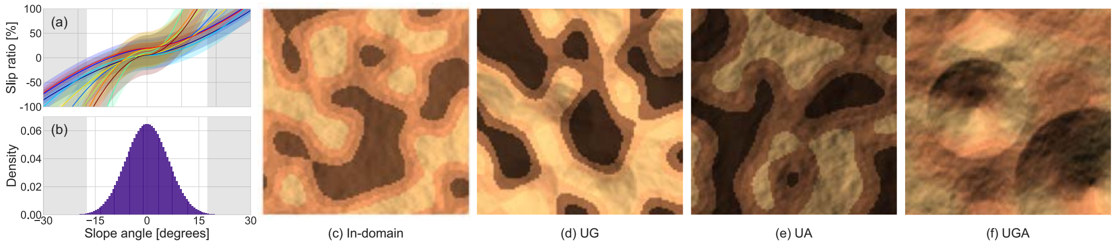


Fig. 5.2: Dataset visualizations. (a) Ten latent slip functions associated with terrain classes, with shaded areas indicating the 2σ ranges of their additive noises. (b) Distribution of terrain slope angles in the training subset. The gray shading in (a) and (b) indicates the sampling range of the crater slopes in (d) and (f). (c)-(f) Example color maps for the in-domain, unfamiliar geometry, unfamiliar appearance, and unfamiliar geometry and appearance test subsets.

prediction.

5.5.1 Dataset

We create a synthetic dataset for path-planning problems, emphasizing the challenge of train–test domain shifts. Thus, we prepare training and validation subsets as in-domain data for training the DPT models and four types of test subsets as out-of-domain data for evaluating navigation performance in unfamiliar environments. Each map instance is a 96×96 grid with a resolution of 1 m per grid, providing color and height maps as inputs for navigation and a ground-truth slip map for model training and traverse simulation during testing. The data generation process is as follows:

Overall data generation

We define 10 terrain classes in the dataset as hidden information. Each class c is associated with the nonlinear stochastic slip function $f_c(\phi) + \epsilon_c$, where $\epsilon_c \sim \mathcal{N}(0, \sigma_c^2)$, as shown in Fig. 5.2a, and appearance color I_c. Subsequently, we create *map templates* by generating random rough terrain geometries using fractal terrain modeling [70]. From each map template, we generate multiple map instances by randomly assigning a terrain class to each node or pixel, along with its associated slip function and color. The assignment is performed using random Perlin noise images to form several class clusters on each map. Slip maps are generated by sampling the slips from the assigned slip functions. For the color maps, shading is applied to obtain more realistic observation imagery (Fig. 5.2c-f). These shaded color maps are computed as $I_v = I_{c(v)} \max(\boldsymbol{n}_v^T \boldsymbol{\ell}, \epsilon)$

for each node v, where $\boldsymbol{n}_v$ is a surface normal vector, $\boldsymbol{\ell}$ is a random lighting direction vector, and ϵ is a small positive constant representing the ambient light level. For the training, validation, and test subsets, 20,000, 5,000, and 400 map instances are created from 100, 25, and 40 map templates, respectively.

In-domain data generation

We control two design factors for data generation to distinguish between in-domain and out-of-domain (OOD) data. The first factor is the steepness of the geometry in the map templates. The other factor is the lighting angle during shading. Specifically, the light direction vector is sampled in spherical coordinates with an azimuth angle $\psi \sim U(0, 2\pi)$ and a vertical component $z \sim U(z_{\min}, z_{\max})$, where $z_{\min}$ and $z_{\max}$ control the elevation angle range. As in-domain data for training and validation, we adopt fractal terrain geometries, resulting in moderate steepness, with slope angles mostly ranging within $\pm 17°$ (see Fig. 5.2b). Additionally, we set the lighting parameters $\{z_{\min}, z_{\max}\}$ to $\{0.8, 1.0\}$.

Out-of-domain test data generation

For the OOD geometry setting, we introduce steeper crater-like geometries to the fractal-terrain geometries (Fig. 5.2d and f), with their steepness randomly sampled from the range of 17.5°–30° (see the shaded regions in Fig. 5.2a and b). For the OOD appearance setting, we set the lighting parameters $\{z_{\min}, z_{\max}\}$ to $\{0.3, 0.5\}$. By combining these two OOD factors, we create the following four test subsets:

- **In-domain subset** (Fig. 5.2c) simply follows the same data distributions as the training and validation subsets for evaluating baseline performance under familiar environmental conditions.
- **Unfamiliar geometry (UG) subset** (Fig. 5.2d) contains OOD crater-like geometries with steeper slopes.
- **Unfamiliar appearance (UA) subset** (Fig. 5.2e) contains OOD appearances with darker shading due to lower lighting angles.
- **Unfamiliar geometry and appearance (UGA) subset** (Fig. 5.2f) combines both OOD factors.

Each test subset contains 100 map instances generated from 10 map templates. Also, we prepare 40 maps, similar to the test subsets but 10 times smaller in size, for tuning the hyperparameters of planning and adaptation algorithms.

One could consider using real-world data directly; however, real-world data often remain constrained in variety, potentially resulting in biased verification or mere demonstration of the proposed approach. By contrast, the datasets introduced in this section are more extensive, covering a wide range of scenarios including OOD cases. As a result, these datasets enable more thorough evaluations compared to using a single real-world dataset.

5.5.2 Implementation Details

For DPT, we use $M = 10$ networks. Each network f_{θ_m} adopts U-Net [88] with ResNet-18 [89] as the CNN. We also apply vector embedding to slope angles before feeding them into the MLP. During pre-training, each network f_{θ_m} is trained on the training subset for 100 epochs using the Adam optimizer [90] with a learning rate of 10^{-3} and batch size of 256. The model weights that produce the lowest validation loss are used for evaluating the test subsets. Planning is performed using an A* search [117], which provides an optimal solution for grid maps if one exists. Both planning and execution assume a velocity u_{ref} of 0.1 m/s. In each adaptation step, backpropagation runs for 10 iterations with a learning rate of 10^{-3} for each f_{θ_m} to exploit the lowest-loss models for subsequent replanning.

5.5.3 Experimental Setup

For each instance, the planners search for a path from the start (16 m, 16 m) to the goal (80 m, 80 m). After finding a solution, the rover navigates the map while receiving noisy edgewise slips s_e. A solution is considered successful if the rover never observes $s_e \geq 1$ (rover immobilization) or $s_e \leq -1$ (dyscontrol situation) along the path toward the goal. We employ the following three performance metrics:

- **Success rate** (Suc) [%] denotes the percentage of problem instances with successful solutions out of the total number of instances provided.

- **Total time** (T_{total}) [min] measures the path efficiency as total traversing time per successful path.

Table 5.1: Quantitative path planning results on 400 problem instances from the four test subsets

	GT		In-domain		
	Slp	Cls	Suc↑	$T_{\text{total}}\downarrow$	$s_{\max}\downarrow$
Seg+EV [1]	✓	✓	**99**	**24.1** ± 0.9	48.5 ± 13.3
MoE+CVaR [2]	✓	✓	**99**	24.4 ± 1.1	**45.6** ± 12.2
Ours	✓		98	24.6 ± 1.2	46.7 ± 12.9
Ours w/ DA	✓		98	24.6 ± 1.3	47.9 ± 13.3
	GT		**UG**		
	Slp	Cls	Suc↑	$T_{\text{total}}\downarrow$	$s_{\max}\downarrow$
Seg+EV [1]	✓	✓	93	**24.9** ± 1.5	61.1 ± 17.6
MoE+CVaR [2]	✓	✓	**94**	25.1 ± 1.4	57.2 ± 16.8
Ours	✓		93	25.5 ± 1.5	**56.4** ± 14.0
Ours w/ DA	✓		91	25.4 ± 1.6	56.6 ± 15.8
	GT		**UA**		
	Slp	Cls	Suc↑	$T_{\text{total}}\downarrow$	$s_{\max}\downarrow$
Seg+EV [1]	✓	✓	84	**25.1** ± 1.1	64.4 ± 19.7
MoE+CVaR [2]	✓	✓	87	25.5 ± 1.4	65.7 ± 18.5
Ours	✓		98	25.2 ± 1.1	57.2 ± 12.9
Ours w/ DA	✓		**99**	25.2 ± 1.4	**55.2** ± 13.9
	GT		**UGA**		
	Slp	Cls	Suc↑	$T_{\text{total}}\downarrow$	$s_{\max}\downarrow$
Seg+EV [1]	✓	✓	66	29.8 ± 34.6	77.2 ± 20.9
MoE+CVaR [2]	✓	✓	69	**26.2** ± 1.9	77.6 ± 18.8
Ours	✓		87	26.4 ± 2.5	61.8 ± 18.4
Ours w/ DA	✓		**90**	26.5 ± 3.0	**58.6** ± 17.1

Suc, T_{total}, and $s_{\max}$ denote success rate [%], total time [min], and maximum slip [%], respectively. Suc and $s_{\max}$ indicate path safety, and T_{total} reflects path efficiency. T_{total} and $s_{\max}$ are shown as mean ± standard deviation. GT indicates the use of ground-truth slip (Slp) and class (Cls) annotations.

- **Maximum slip** ($s_{\max}$) [%] evaluates path safety by calculating $100 \times \max(\boldsymbol{s_e})$ post-execution, including failure cases. $\boldsymbol{s_e}$ denotes the experienced slips; a lower $s_{\max}$ indicates a safer traverse.

We run our method using $\lambda = 2.0$ in (5.8) with and without domain adaptation (DA) (described in Section 5.4.3). We compare different prediction models and risk evaluation methods to validate the proposed approach. As a baseline, we adopt a segmentation-based planner (Seg+EV) [1] that selects single class-associated Gaussian processes based on terrain classification and simply calculates the expected values (EVs) of the GPs for planning costs, ignoring prediction uncertainties. We also evaluate a MoE-based risk-aware planner (MoE+CVaR) [2], which integrates multiple GPs with terrain classification likelihoods and calculates risk-aware planning costs with CVaR using $\alpha = 0.9$. These methods use the same CNN architecture as ours for terrain classification. However, the training of CNNs and class-associated GPs relies on additional ground-truth (GT) class annotations from the training subset (see GT in Table 5.1).

5.5.4 Results and Discussions

Quantitative Comparison

Table 5.1 summarizes the path-planning performance across the four test subsets. Overall, our approach performs comparably to the baseline methods for the in-domain and UG subsets and significantly outperforms them for the UA and UGA subsets, despite using only slip annotations for training. Comparisons of our results with each baseline and the impact of DA are discussed below.

vs. Seg+EV: The Seg+EV-based planner [1] tends to prioritize path efficiency over safety because its EV-based planning costs ignore prediction uncertainties. While this tendency results in the lowest T_{total} for all subsets but UGA, our approach maintains a comparably low T_{total} with much higher safety for the GA and UGA subsets. These benefits stem from our time-based cost evaluation using uncertainty-aware slip estimation in (5.8), which balances traversability and travel time and is beneficial for successful planetary explorations that require both safety and efficiency.

vs. MoE+CVaR: Although the MoE+CVaR-based planner [2] also incorporates uncertainties in slip predictions for risk-aware planning, it experiences low success rates for the UA and the most challenging UGA subsets. We found that the darker shaded regions in these subsets often cause misclassifications by the terrain classifier in the MoE slip model. Conversely, our DPT fuses color-based and geometry-based features to predict slip distributions end to end, effectively disambiguating terrain colors and shading effects to improve predictions.

Table 5.2: Domain adaptation effects for slip prediction

	In-domain	UG	UA	UGA
Before DA	**8.7** ± 1.6	12.3 ± 2.6	12.6 ± 1.2	16.3 ± 2.3
After DA	**8.7** ± 1.5	**11.7** ± 1.8	**11.1** ± 1.3	**14.3** ± 2.1

The 100 times scaled mean absolute errors of the slip predictions by the proposed DPT model before and after the domain adaptation.

Impact of Domain Adaptation: DA successfully improves the success rates and $s_{\max}$ scores for the UA and the most challenging UGA subsets. For the in-domain subset, the DA maintains the same performance as that before adaptation, as the new in-situ samples fall within the pretraining domain. However, for the UG subset, we observe a slight decrease in the success rate. This subset poses the difficulty of adapting and correctly extrapolating the learned slip functions to OOD slope angles (*i.e.*, $|\phi| > 17°$) given a few OOD samples. Because the current architecture of our slip model does not strictly follow the monotonicity inherent in latent slip functions $s = f(\phi)$, such an extrapolation through few-shot adaptation may be challenging. Incorporating monotonicity into the DPT slip model may improve its performance and adaptation stability. Still, we found that DA consistently improves the average slip prediction accuracy across the four test subsets. The results are provided in Table 5.2, comparing the mean absolute errors of the slip predictions for entire map regions before and after test-time adaptation.

Qualitative Discussion

Fig. 5.3 summarizes rover navigation results using the proposed unified framework in a UGA subset problem instance, including input maps for traversability prediction (Fig. 5.3a), along with ground truth annotations for model training (Fig. 5.3b). Fig. 5.3c shows the probabilistic slip predictions for rightward movements, comparing the results before and after DA.

In the results of the pre-trained DPT model before adaptation (top row), the absolute error map shows robust slip predictions, except in OOD terrain areas, such as crater-type terrain with steep inclinations and intense shading. The errors in these areas are highly correlated with the intense uncertainties in the total uncertainty map. This capability provides a comprehensive assessment of the predictions and their confidence, helping to identify both hazardous and unfamiliar terrain conditions. Uncertainty

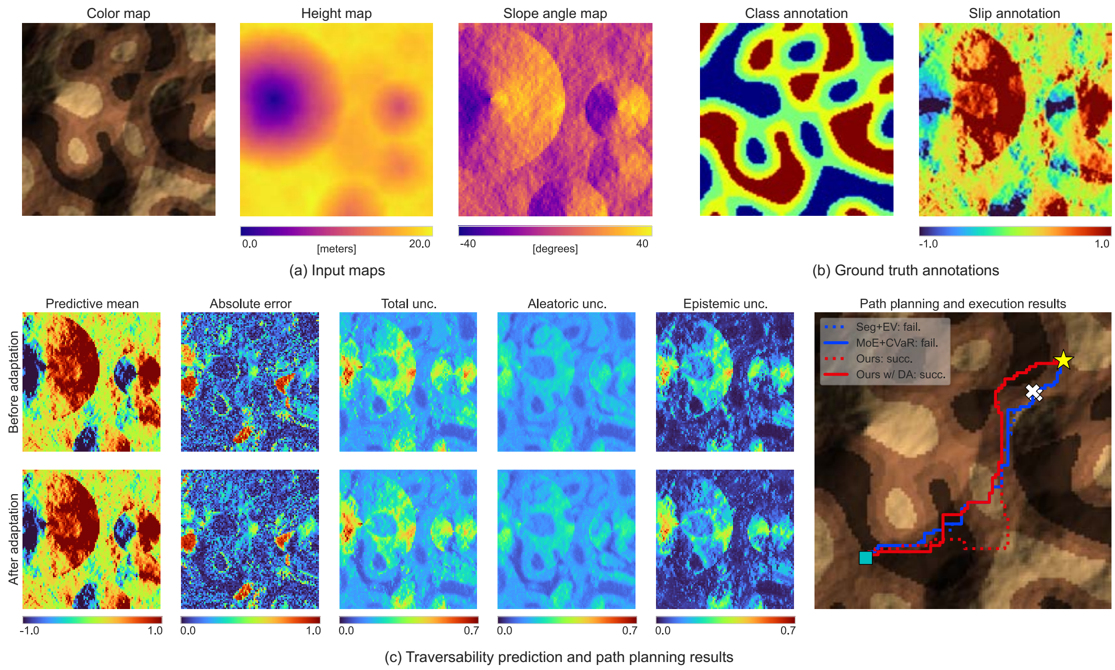


Fig. 5.3: Rover navigation using the proposed unified framework compared with methods [1, 2] for an unfamiliar geometry and appearance subset problem instance. (a) Color, height, and slope maps as model inputs for rightward movements. (b) Ground truth annotations, with class annotations used for training terrain classifiers, while DPT only requires slip annotations. (c) Traversability predictions and path planning results. Left: Slip predictions before (top row) and after (bottom row) domain adaptation, displaying absolute errors with total, aleatoric, and epistemic uncertainties. Right: Path planning and execution results. The cyan square and yellow star indicate the start and goal locations, respectively. White crosses mark locations where rovers failed to traverse.

decomposition further reveals epistemic components around OOD areas and aleatoric components aligned with terrain class patterns owing to class-dependent noise in the latent slip functions.

In the results of the adapted DPT model (bottom row), we observe reduced errors for OOD regions while maintaining the overall prediction trends. This robust adaptation is enabled by our adaptation model, which tunes only the scale and bias parameters.

On the right side of Fig. 5.3, we compare the planned paths obtained using our method and baseline methods. Our method achieves more robust path planning than the other methods by integrating these well-calibrated uncertainty estimates into traversability evaluation. The MoE+CVaR-based planner also exploits this uncertainty. However, its misallocated class probabilities cause the same traverse failure as

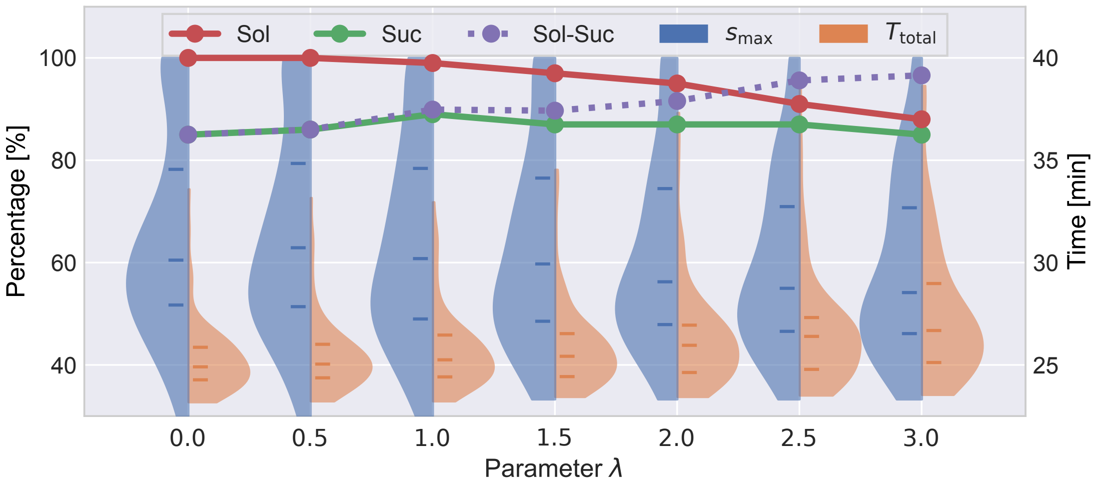


Fig. 5.4: Parameter study for the unfamiliar geometry and appearance subset, showing the variations of five metrics with varying λ. The left percentile y-axis represents Sol, Suc, Sol-Suc, and $s_{\max}$, while the right y-axis represents T_{total}. Blue and orange violin plots represent the distribution of $s_{\max}$ and T_{total}, respectively, with lines marking the 25th, 50th, and 75th percentiles.

that caused by the Seg+EV-based planner.

Parameter Study

We assess the impact of uncertainty exploitation for the UGA subset, by running our algorithm using $\lambda \in \{0.0, 0.5, \dots, 3.0\}$ without DA. We introduce additional performance metrics: *solved rate* (Sol) [%] as the percentage of problem instances for which a planner finds solutions and *solved success rate* (Sol-Suc) [%] as the success rate within solved cases. Fig. 5.4 illustrates how λ controls the balance between safety and efficiency: higher λ values guide rovers to less slippery paths, while lower values allow more aggressive maneuvers to reduce travel time. For safer navigation, higher λ is preferable. Although increasing λ results in lower solution rates, it improves solved success rates, suggesting that planning failures effectively predict serious immobilization risks in the environment. The optimal λ depends on the mission specifics and environment, but statistical considerations offer general guidelines. Accounting for prediction errors within the 2σ range (95% confidence interval) using $\lambda = 2$ balances the safety and efficiency across various scenarios.

Limitations and Possible Extensions

The DPT architecture lacks monotonicity constraints between slope angles and slip values, which degrades performance in unfamiliar geometry scenarios. Monotonic neural networks [118] could provide more reliable slip predictions for OOD slope angles. Domain adaptation also shows limited effectiveness across OOD scenarios, as shown in Table 5.2, which suggests the need for better sample exploration methods to reduce epistemic uncertainty. Active domain adaptation [80] could leverage using epistemic uncertainty components of DPT as cues for informative samples for training, although safe data collection in hazardous terrain remains challenging.

5.6 Summary

We have introduced a unified learning and planning framework for navigating rovers on deformable terrain in celestial environments, focusing on ML-based traversability prediction and uncertainty handling. To test the effectiveness of our method, we created a synthetic dataset that highlights the challenge of train-test domain shifts in traversability prediction. Extensive simulation experiments using this dataset have demonstrated that our approach enables safe rover traverse with improved success rates compared to existing approaches, even under unfamiliar geometry and appearance conditions. Test-time domain adaptation through the end-to-end architecture of our ML model also contributes to successful navigation by continuously reducing prediction errors as the rovers traverse. Future work will explore an active learning strategy to reduce uncertainties in the environment efficiently through domain adaptation. In addition, we plan to incorporate real-world data (*e.g.*, the Martian terrain data [91]) to examine scenarios with unfamiliar environmental features, analogous to those in our unfamiliar geometry and appearance dataset. This will further verify the effectiveness of our approach and highlight the thoroughness of the datasets prepared in this study.

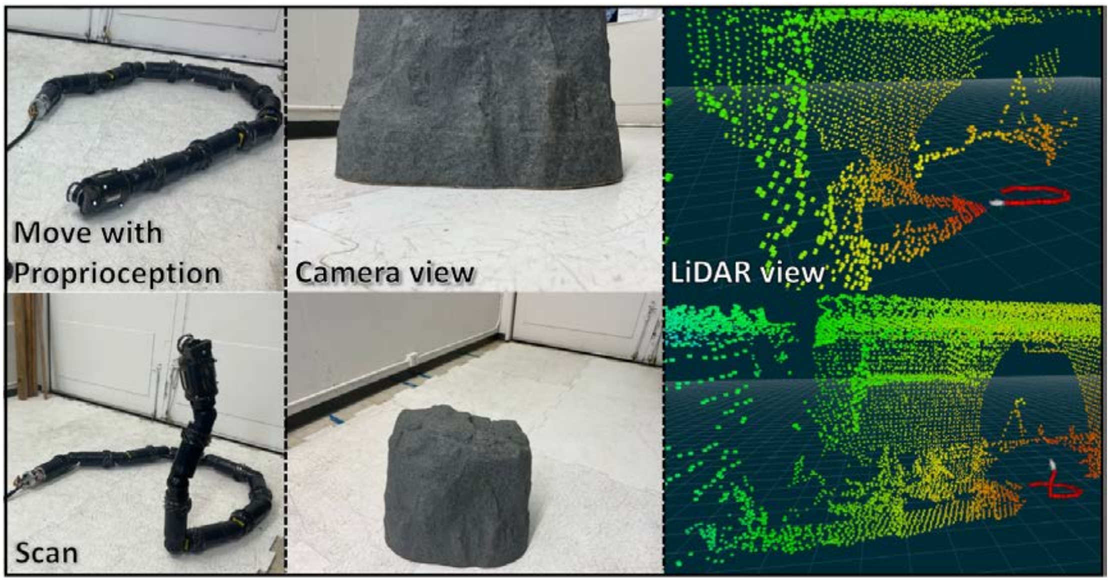
Chapter 6

Integrated Task and Motion Planning for State Transition Uncertainty

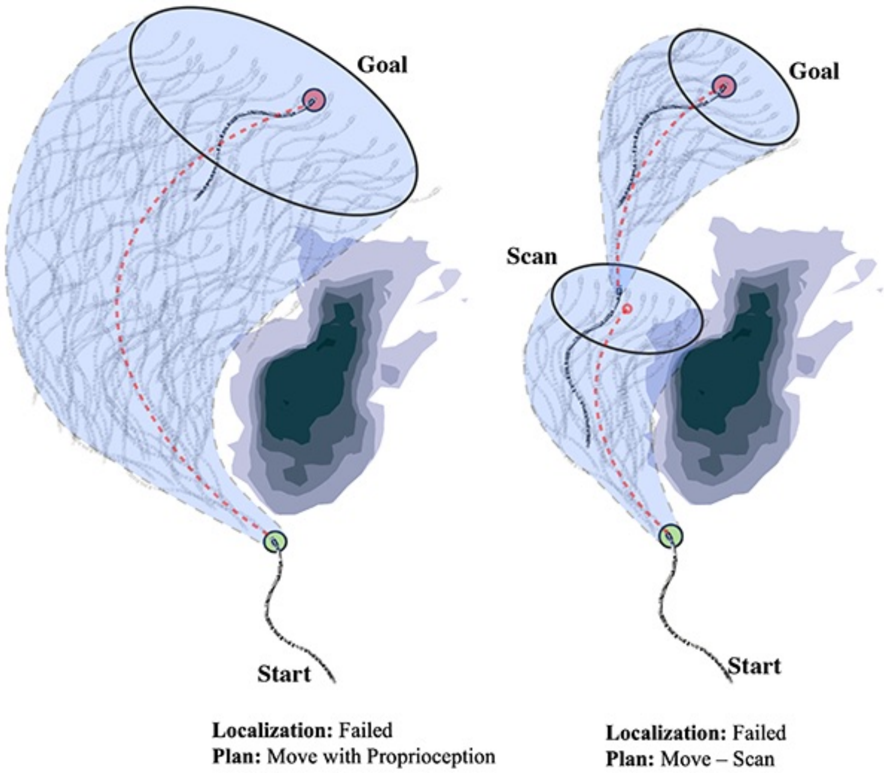
6.1 Introduction

Snake robots have versatile mobility for exploring extreme environments due to their reconfigurable body structure. These robots are applicable for both terrestrial and space missions, from search and rescue in confined spaces to exploring ice worlds like Enceladus or Europa, navigating diverse terrains from vast surfaces to deep crevasses. However, snake robots struggle with the proximity of their sensor payload to the ground, which increases the complexity of simultaneous localization and mapping. In complex terrain environments, the proximity of sensors to the ground leads to substantial occlusion by natural obstacles and terrain undulations and accelerates their motion in image space, raising the risk of localization failures. A restricted view from low-mounted LiDAR also increases the risk of front-end failures in algorithms such as iterative closest point [119]. The rapid motion of features in the robot's view further induces sensor data uncertainty and distortion. Maintaining an upright scan posture, with the robot's front raised, offers a solution, but also consumes energy and limits locomotion capability (Fig. 6.1a). To ensure safe and efficient mobility under uncertainty, snake robots require simultaneous decisions for *where to move* and *when to scan* to reveal unexplored regions.

To this end, we propose *Blind-motion with Intermittently Scheduled Scans (BLISS)* to



(a) Proprioception enables blind movement while scanning collects sensor data. Ground-level sensors increase localization errors. Scanning improves localization and obstacle detection by extending EELS' field of view.



(b) Tradeoff between longer paths with higher obstacle clearance but less frequent scans (left) and shorter paths with lower obstacle clearance but more frequent scans (right) to manage collision risk.

Fig. 6.1: Blind-motion with Intermittently Scheduled Scans (BLISS).

achieve resiliency to localization failures by reducing reliance on exteroceptive feedback during motion while scheduling scanning actions for periodic localization updates. As illustrated in Fig. 6.1b, BLISS tackles an integrated task and motion planning (TAMP) problem to balance the tradeoff between safety and efficiency: Frequent scanning actions ensure safe navigation but consume additional time for accurate localization, while reduced scanning enables faster, continuous motion at the cost of increased collision risk. However, formulating BLISS-TAMP naturally leads to a hybrid partially observable Markov decision process (HPOMDP), a model that encodes both discrete task decisions and continuous motion variables under partial observability. HPOMDPs are known to be computationally challenging, especially for long-horizon missions or complex environments. A direct approach often becomes intractable due to the large number of decision variables and the complexity of handling partial observability over extended time horizons.

We address this challenge through two complementary approaches. First, we reformulate the HPOMDP as a convex mixed integer linear programming (MILP) framework, which ensures computational efficiency through its convex structure. Second, we introduce a novel *continuous-time* planning formulation that eliminates explicit time discretization, which drastically reduces the dimensionality of the optimization problem with fewer variables and constraints. This combined approach maintains all dynamics and safety constraints traditionally formulated in discrete-time systems. The computational tractability is achieved through both convex MILP optimization and reduced dimensionality in the continuous-time domain. Although we focus on snake robots, our method applies broadly to any robotic system requiring integrated planning under uncertainty, proprioceptive mobility, and periodic information gathering.

6.2 Related Work

6.2.1 TAMP Without Uncertainty

Early work in Task and Motion Planning (TAMP) often treats task planning and motion planning as separate, sequential processes. The task planner generates high-level symbolic goals, while the motion planner handles geometric path feasibility [120–127]. While some research has explored integrating these two levels [128–134], these methods generally assume deterministic environments or ignore the significant impact of uncertainty on plan quality.

6.2.2 Incorporating Uncertainty into TAMP

To navigate more realistic scenarios, TAMP must account for imperfect sensing or incomplete knowledge. However, introducing uncertainty into TAMP drastically increases complexity, especially when the agent must plan over discrete actions and continuous states under partial observability. Approaches that incorporate uncertainty usually do so within a discrete task planning framework, where conditions and contingencies are encoded as branches [135,136], or through probabilistic task planning [137]. Despite these advances, the complexity of enumerating possible outcomes and updating beliefs quickly becomes intractable, even in relatively simple domains.

6.2.3 Risk-aware vs. Robust Formulations

A key challenge in TAMP under uncertainty is how to handle the probabilistic nature of state evolution and observations. Risk-aware approaches, including chance-constrained formulations, have been applied to both HPOMDPs [138,139] and robotic exploration [140, 141]. These methods aim to keep the probability of failure below a specified threshold. In motion planning, there are numerous methods to address probabilistic constraints, from sampling-based [142, 143] to optimization-based [144, 145] techniques. However, these methods rely on complex probability distributions to characterize the uncertainty evolution, which substantially increases the computational complexity as maintaining and updating these distributions requires high-dimensional belief space representations. As a result, ensuring risk constraints over longer horizons or in more complicated settings becomes computationally demanding.

In contrast, robust approaches guarantee safety and feasibility under all realizations of uncertainty within predefined bounded sets. They replace probabilistic distributions with deterministic uncertainty sets, thereby avoiding the complexity of belief updates. While this can yield more tractable formulations and facilitate integrating task and motion planning, the resulting optimization problems remain inherently NP-hard, posing challenges for efficient large-scale solutions.

6.2.4 BLISS: From Risk-aware to a Continuous-time Robust TAMP

Our previous work [146] introduced BLISS, integrating blind locomotion and scanning to mitigate localization uncertainty. This approach formulated BLISS-TAMP as a chance-constrained HPOMDP (CC-HPOMDP) and translated it into a convex Mixed Integer Linear Program (MILP) by relying on belief MDP approximations and quadratic covariance growth models. While effective for simple scenarios, the discrete-time, risk-aware formulation encountered scaling issues in complex environments, as each planning step introduced additional optimization variables and constraints.

6.2.5 Our Contribution

In this chapter, we advance the state of the art in TAMP under uncertainty by adopting a continuous-time formulation. We eliminate the need for discrete time steps and thereby drastically reduce both integer and continuous variables in the optimization problem, with the reduction in integer variables particularly beneficial for MILP computational complexity. As a result, the corresponding MILP becomes more tractable, offering substantially improved computational efficiency (see Table 6.1). While chance-constrained formulations are appealing for their explicit consideration of probabilistic uncertainty, they often lead to nonconvex optimization problems that are challenging to solve efficiently. As an initial step toward tractable planning under uncertainty, we employ a robust perspective that guarantees safe navigation under worst-case conditions, serving as a stepping stone for future risk-aware formulations. As Table 6.1 highlights, our approach stands out by providing integrated TAMP, robust safety assurances, and substantial computational gains in a continuous-time framework.

Our contributions can be summarized as follows:

- Introduction of BLISS for systems needing intermittent scanning and blind locomotion to address localization challenges in extreme environments.
- A continuous-time TAMP formulation that removes the reliance on discrete time steps, reducing optimization complexity and enabling real-time or near-real-time planning solutions.
- A robust perspective that avoids probability distributions, ensuring guaranteed safety and tractability even in complex, long-horizon missions.

Table 6.1: Representative related work landscape

Work	Integrated TAMP	Under Uncertainty	Risk-aware Constraints	Robust Constraints	Compute Efficiency
[120–127]					
[128–134]	✓				
[135–137]	✓	✓			
[138–141]		✓	✓		
[146]	✓	✓	✓		✓
This work	✓	✓		✓	✓✓

6.3 Problem Formulation

In this section, we formulate the BLISS-TAMP problem for long-range navigation under uncertainty. First, we describe the overall objective of BLISS-TAMP and its relevance to snake robots operating in extreme terrains. We then formalize the problem as an HPOMDP, highlighting the interplay between discrete actions (move or scan) and continuous spatial decision variables.

6.3.1 BLISS-TAMP for Long-range Navigation

Robust perception and localization for snake robots remain an open problem. The use of snake robot reconfigurability for scanning behaviors was first explored in [147], where the authors focused on hardware and gait design with limited onboard autonomy. We leverage the EELS robot's scanning capability for long-range localization and obstacle detection, combined with blind proprioception-only mobility for navigation, to be resilient against localization failures when exteroceptive sensors are near the ground.

We propose a long-range TAMP framework, referred to as BLISS, that considers uncertainty in obstacle collision avoidance in extreme terrains. Our planner approximates the robot as a point mass with linear dynamics, generating a low-resolution 2.5-D motion path augmented with scheduled scanning actions. These scans help manage uncertainty in localization and ensure safety against worst-case realizations of disturbances. A short-range planner and controllers then track this long-range solution while accounting for the robot's high DoF and non-linear dynamics [148]. Fig. 6.2 illustrates the layered autonomy architecture. Further details on the robot and associated autonomy components can be found in [13, 149].

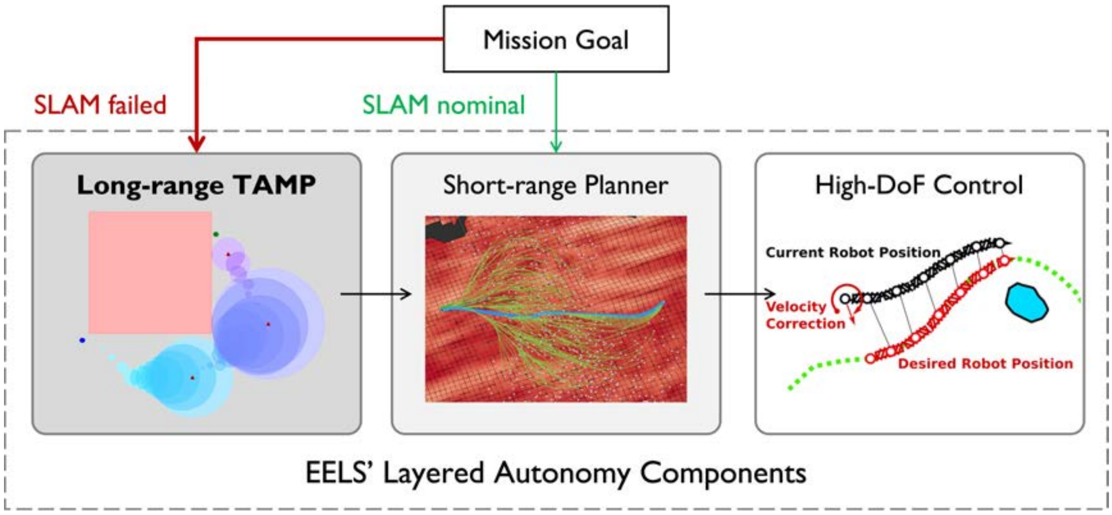


Fig. 6.2: EELS' layered autonomy architecture: from Long-range TAMP (this work), Short-range planner, to High-DoF controllers.

6.3.2 HPOMDP Representation of BLISS-TAMP under Uncertainty

The BLISS-TAMP problem involves a robot (the agent) that must reach a goal region while avoiding obstacles and managing sensor uncertainty. The problem is hybrid because the agent chooses both:

- Discrete actions $a \in \mathcal{A}$, such as $\{Move, Scan\}$.
- Continuous control inputs $u \in \mathbb{R}^{N_u}$, typically velocities in the x, y plane.

Moreover, partial observability arises due to imperfect sensors and limited visibility. The agent does not have direct access to the true state $X \in \mathbb{R}^{N_X}$ but receives measurements corrupted by noise, and when scanning, it updates its belief or uncertainty set.

In a probabilistic setting, this formulation aligns with HPOMDP. The state transitions incorporate both continuous dynamics and discrete mode changes. Observations depend on the actions and sensor noise models. Traditionally, such problems would require belief updates over probability distributions, often leading to intractable *hyperbeliefs* (Fig. 6.3).

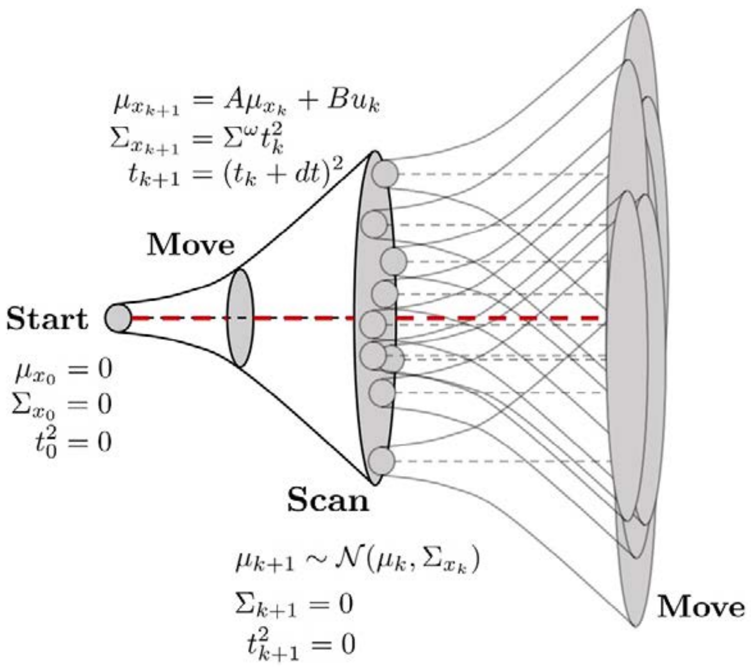


Fig. 6.3: BLISS-TAMP requires belief updates over probability distributions due to the dependence on future observations. The updating process leads to hyperbeliefs, where each belief state itself becomes a distribution, which fundamentally increases the complexity of planning under uncertainty.

6.3.3 From Risk-aware to Robust HPOMDP Formulation

Previous formulations of BLISS-TAMP [146] used a Chance-Constrained HPOMDP (CC-HPOMDP) approach, where uncertainties were modeled probabilistically. Actions and observations were integrated into a belief-space planner that balanced risk by explicitly managing the probability of constraint violation via a risk bound Δ.

In this work, we shift from a risk-aware, probability-based framework to a robust, deterministic set-based one. Instead of representing uncertainty as a probability distribution and imposing chance constraints, we define bounded uncertainty sets. That is, after each action (move or scan), the agent's true state resides in a known compact set around the nominal planned state. This robust interpretation simplifies planning:

- **No Hyperbeliefs:** Without probability distributions, we avoid tracking belief distributions over belief distributions. Instead, we maintain and plan over sets of possible states.
- **Worst-case Guarantees:** By ensuring that the entire uncertainty set remains within safe bounds, we achieve a guaranteed collision-free solution under worst-

case disturbances.

While this robust approach omits explicit probabilistic guarantees, it provides a more tractable problem formulation. The resulting problem can still be considered a form of HPOMDP under uncertainty but with robust set-based uncertainty rather than stochastic uncertainty. In the next section, we introduce a continuous-time formulation that further reduces computational complexity and enables MILP solutions to robust BLISS-TAMP.

6.4 Approach: Continuous-time Robust TAMP

This section presents the continuous-time formulation of our robust TAMP problem. We first describe how to model deterministic motion planning without uncertainty in a continuous-time setting. Building on this foundation, we then incorporate uncertainty and robust constraints to achieve a guaranteed safe solution under worst-case conditions.

6.4.1 Continuous-Time Deterministic Motion Planning

Conventional planning approaches typically discretize time into fixed intervals, introducing a large number of decision variables and thus increasing computational complexity. In contrast, we consider a continuous-time representation of the planning problem. By avoiding fixed-time discretization, we can reduce the number of time steps and their corresponding decision variables, which leads to improved computational tractability.

Decision Variables and Time Parameterization

We partition the planning horizon into N segments, each associated with a continuous time interval. For the i-th segment, we define the start and end times:

$$t_{\text{start}_i}, \quad t_{\text{end}_i}, \quad \Delta T_i = t_{\text{end}_i} - t_{\text{start}_i}, \tag{6.1}$$

where $t_{\text{start}_0} = 0$ and $t_{\text{start}_{i+1}} = t_{\text{end}_i}$ ensure continuity of time. The variable ΔT_i is a continuous decision variable representing the duration of the i-th segment, and is not predetermined by a time grid.

We represent the robot's configuration at the i-th segment by a state vector $X_i =$

$[x_i, y_i]^T \in \mathbb{R}^2$, where x_i and y_i denote the robot's position in a 2D plane. The state increment $\Delta X_i = [\Delta x_i, \Delta y_i]^T$ relates consecutive states as:

$$X_{i+1} = X_i + \Delta X_i. \tag{6.2}$$

Here, ΔX_i is another continuous decision variable determining the robot's displacement during segment i.

Unlike discrete-time methods that fix time steps in advance, our approach directly optimizes ΔT_i. This flexibility allows the planner to adapt time allocation to the complexity of the environment or the required precision.

Objective and Constraints

We consider an objective to minimize total travel time:

$$\min \sum_{i=1}^{N} \Delta T_i, \tag{6.3}$$

subject to a time budget:

$$\sum_{i=1}^{N} \Delta T_i \leq T_{\max}, \tag{6.4}$$

where $T_{\max}$ is a given upper bound on the total allowed mission time.

To ensure kinematic feasibility, we impose a maximum velocity $v_{\max}$. Let $\Delta T_i^{\text{Move}} \leq \Delta T_i$ represent the portion of the segment duration spent moving (the remainder could, for example, be idle time). The robot must not exceed $v_{\max}$:

$$|\Delta X_{i,j}| \leq \Delta T_i^{\text{Move}} \cdot v_{\max}, \tag{6.5}$$

where $\Delta X_{i,j}$ denotes the j-th component of the displacement vector $\Delta X_i \in \mathbb{R}^2$.

In this deterministic setting, we assume perfect knowledge of the robot state and environment. Scanning actions, if any, either occur instantaneously or have fixed known durations that do not alter the fundamental dynamic constraints. For now, we focus solely on motion feasibility.

Deterministic Safety Constraints

To ensure collision-free navigation, we model obstacles as polygonal constraints defined by half-planes. Suppose an obstacle or safe corridor is represented by M half-planes:

$$\text{safe}^{\text{obs}}_{s_k} = \{X \in \mathbb{R}^2 \mid a_k^{\text{obs}^T} X \geq b_k^{\text{obs}}\}, \quad k = 1, \ldots, M, \tag{6.6}$$

where $a_k^{\text{obs}} \in \mathbb{R}^2$ and $b_k^{\text{obs}} \in \mathbb{R}$ define the k-th half-plane.

To ensure that the trajectory remains safe, we must guarantee that for each segment i, the pair of endpoints (X_i, X_{i+1}) jointly resides within at least one of these half-planes. Formally:

$$\bigvee_{k=1}^{M} \left(a_k^{\text{obs}^T} X_i \geq b_k^{\text{obs}} \wedge a_k^{\text{obs}^T} X_{i+1} \geq b_k^{\text{obs}} \right). \tag{6.7}$$

Since we do not discretize time, ensuring safety at the endpoints X_i and X_{i+1} of each segment is sufficient to guarantee that the straight-line path between them is collision-free (due to convexity of half-planes).

To encode the logical OR in (6.7) as linear constraints, we introduce binary variables $y_{i,k}^{\text{obs}} \in \{0, 1\}$, one for each half-plane and segment. Let $M_{\text{big}} > 0$ be a sufficiently large constant (big-M). The constraints become:

$$a_k^{\text{obs}^T} X_i \geq b_k^{\text{obs}} - M_{\text{big}}(1 - y_{i,k}^{\text{obs}}), \tag{6.8}$$

$$a_k^{\text{obs}^T} X_{i+1} \geq b_k^{\text{obs}} - M_{\text{big}}(1 - y_{i,k}^{\text{obs}}), \tag{6.9}$$

$$\sum_{k=1}^{M} y_{i,k}^{\text{obs}} \geq 1, \quad \forall i. \tag{6.10}$$

If $y_{i,k}^{\text{obs}} = 1$, then X_i and X_{i+1} must satisfy $a_k^{\text{obs}^T} X \geq b_k^{\text{obs}}$. If $y_{i,k}^{\text{obs}} = 0$, the constraint is relaxed by the big-M term, effectively disabling that half-plane. The final inequality ensures at least one half-plane is active per segment, thus preserving safety.

Mixed Integer Linear Programming Formulation

Combining these elements—continuous-time parameterization, dynamic feasibility, and logical obstacle avoidance constraints—results in a Mixed Integer Linear Program (MILP):

$$\min_{X_i,\Delta X_i,\Delta T_i,y_{i,k}^{\text{obs}}} \quad \sum_{i=1}^{N} \Delta T_i \tag{6.11}$$

$$\text{s.t.} \quad X_{i+1} = X_i + \Delta X_i, \qquad i = 1, \ldots, N-1, \tag{6.12}$$

$$\sum_{i=1}^{N} \Delta T_i \leq T_{\max}, \tag{6.13}$$

$$|\Delta X_{i,j}| \leq \Delta T_i^{\text{Move}} \cdot v_{\max}, \qquad \forall i, j \in \{1, 2\}, \tag{6.14}$$

$$a_k^{\text{obs}^T} X_i \geq b_k^{\text{obs}} - M_{\text{big}}(1 - y_{i,k}^{\text{obs}}), \qquad \forall i, k, \tag{6.15}$$

$$a_k^{\text{obs}^T} X_{i+1} \geq b_k^{\text{obs}} - M_{\text{big}}(1 - y_{i,k}^{\text{obs}}), \qquad \forall i, k, \tag{6.16}$$

$$\sum_{k=1}^{M} y_{i,k}^{\text{obs}} \geq 1, \qquad \forall i, \tag{6.17}$$

$$y_{i,k}^{\text{obs}} \in \{0, 1\}, \qquad \forall i, k. \tag{6.18}$$

Here, $X_i, \Delta X_i, \Delta T_i$ are continuous decision variables representing states, increments, and times, while $y_{i,k}^{\text{obs}}$ are binary variables handling logical constraints for obstacle avoidance. This deterministic continuous-time formulation reduces complexity compared to time-discretized methods, yet still assumes perfect knowledge and no uncertainty.

In the next section, we will extend this framework to incorporate uncertainty, thereby leading to a robust TAMP formulation capable of ensuring safety under worst-case conditions.

6.4.2 Continuous-time Robust TAMP Extension

We now extend the deterministic continuous-time framework to handle uncertainty, which ensures safety against worst-case disturbances. The key idea is to incorporate bounded uncertainty sets into the state definition and modify the constraints so that feasibility holds for all allowable uncertainty realizations.

Uncertainty Modeling and Scanning Actions

Let $\bar{X}_i \in \mathbb{R}^2$ be the nominal planned state at segment i, and let $\delta X_i \in \mathbb{R}^2$ be a disturbance vector representing uncertainty. We assume δX_i belongs to a bounded set:

$$\{\delta X_i \in \mathbb{R}^2 \mid \|\delta X_i\| \leq \epsilon_i\}, \tag{6.19}$$

where $\epsilon_i \geq 0$ is a continuous variable bounding the magnitude of uncertainty at segment i. Thus, the actual state X_i satisfies:

$$X_i = \bar{X}_i \oplus \{\delta X_i \mid \|\delta X_i\| \leq \epsilon_i\}. \tag{6.20}$$

Without additional measures, ϵ_i may grow over time due to localization drift or process noise. To mitigate this, we introduce scanning actions that reduce uncertainty. Let $c_i \in \{0, 1\}$ be a binary decision variable indicating whether a scan occurs during segment i. We partition the segment duration into motion time ΔT_i^{Move} and scanning time ΔT_i^{Scan}:

$$\Delta T_i = \Delta T_i^{\text{Move}} + \Delta T_i^{\text{Scan}}, \tag{6.21}$$

with

$$\Delta T_i^{\text{Scan}} = c_i \Delta T_{\text{Scan}}, \tag{6.22}$$

where $\Delta T_{\text{Scan}} > 0$ is a given constant representing the fixed duration needed for a scan action. If $c_i = 1$, the robot performs scanning and does not move in that segment ($\Delta X_i = 0$).

Uncertainty Propagation and Reduction via Scanning

We model how uncertainty propagates along the robot's trajectory and how it can be reduced through strategic scanning actions. As introduced in Section 6.4.2, each segment i is characterized by a nominal robot state $\bar{X}_i$, an actual, uncertain state X_i, and a total duration ΔT_i that may include both movement and scanning time.

General Disturbance-Based Model. To capture worst-case disturbances, let $W \subset \mathbb{R}^2$ be a compact set representing all possible external disturbances that could arise during movement. A linearized representation of the state evolution under disturbance can be written as

$$X_{i+1} \;=\; X_i \;+\; \Delta X_i \;+\; \Delta T_i^{\text{Move}} \cdot w_i, \quad \forall w_i \in W, \tag{6.23}$$

where ΔX_i is the nominal displacement and ΔT_i^{Move} is the portion of time spent moving. Ensuring robust feasibility (e.g., collision avoidance) then requires all constraints to hold *for every* possible $w_i \in W$. A common approach in robust optimization is to propagate an *uncertainty set* $\mathcal{E}_i$ through (6.23), such that $X_i \in \mathcal{E}_i$ for all i. However, directly encoding set-based propagation in MILP can be computationally demanding.

Proposed Instantiation: Scalar Uncertainty with Scanning. To balance robustness and tractability, we adopt the same scalar uncertainty approach introduced in Section 6.4.2, where the actual state X_i at segment i remains within an ϵ_i-ball around the nominal plan $\bar{X}_i$, *i.e.*, $\|X_i - \bar{X}_i\| \leq \epsilon_i$. Building on this framework, we further approximate the growth of ϵ_i as linear in the movement time ΔT_i^{Move}, and assume that performing a scan localizes the robot sufficiently to reset this uncertainty:

$$\epsilon_{i+1} \;=\; \epsilon_i \;+\; z\sigma_w \, \Delta T_i^{\text{Move}} \;-\; c_i \, \epsilon_i, \quad c_i \in \{0, 1\}. \tag{6.24}$$

Here, $z\sigma_w > 0$ is a parameter modeling the rate of uncertainty growth, and $c_i = 1$ indicates that a scanning action is performed in segment i. As explained in Section 6.4.2, each ΔT_i may include both movement and scanning durations as needed.

Remark: This scalar formulation can be viewed as an instantiation of (6.23) *under norm-bounded disturbances* $\|w_i\| \leq z\sigma_w$ *and the assumption that a scan action effectively resets the accumulated uncertainty. Though idealized, this assumption remains practical in real-world scenarios where scanning significantly reduces localization drift and sensor noise, thereby providing a tractable yet robust planning framework.*

By tracking only the scalar ϵ_i, we achieve a linear formulation for robust obstacle avoidance through a big-M approach, making the overall MILP computationally more manageable.

Robust Obstacle Avoidance Constraints

As in the deterministic case, we model obstacles using polygonal safe corridors defined by M half-planes. Let:

$$\text{safe}_{s_k}^{\text{obs}} = \{X \in \mathbb{R}^2 \mid a_k^{\text{obs}^T} X \geq b_k^{\text{obs}}\}, \quad k = 1, \ldots, M, \tag{6.25}$$

where $a_k^{\text{obs}} \in \mathbb{R}^2$ and $b_k^{\text{obs}} \in \mathbb{R}$ characterize the k-th half-plane constraint.

In the robust setting, the actual state X_i can vary within an ϵ_i-ball around $\bar{X}_i$. To ensure X_i remains safe for all realizations of $\|\delta X_i\| \leq \epsilon_i$, we must enforce:

$$a_k^{\text{obs}^T} \bar{X}_i - \epsilon_i \|a_k^{\text{obs}}\| \geq b_k^{\text{obs}}, \tag{6.26}$$

for any chosen safe half-plane. This ensures that even if X_i is perturbed by up to ϵ_i in any direction, it remains on the safe side of the k-th boundary.

Similar to the deterministic formulation, we require that for each segment i, both endpoints $(\bar{X}_i, \bar{X}_{i+1})$ lie robustly within at least one safe half-plane. Formally:

$$\bigvee_{k=1}^{M} \left(a_k^{\text{obs}^T} \bar{X}_i - \epsilon_i \| a_k^{\text{obs}} \| \geq b_k^{\text{obs}} \ \wedge \ a_k^{\text{obs}^T} \bar{X}_{i+1} - \epsilon_{i+1} \| a_k^{\text{obs}} \| \geq b_k^{\text{obs}} \right). \tag{6.27}$$

To linearize this logical OR, we proceed as in the deterministic case: introduce binary variables $y_{i,k}^{\text{obs}} \in \{0,1\}$ for each half-plane and apply a big-M method. Let $M_{\text{big}} > 0$ be a sufficiently large constant. Then:

$$a_k^{\text{obs}^T} \bar{X}_i - \epsilon_i \| a_k^{\text{obs}} \| \geq b_k^{\text{obs}} - M_{\text{big}}(1 - y_{i,k}^{\text{obs}}), \tag{6.28}$$

$$a_k^{\text{obs}^T} \bar{X}_{i+1} - \epsilon_{i+1} \| a_k^{\text{obs}} \| \geq b_k^{\text{obs}} - M_{\text{big}}(1 - y_{i,k}^{\text{obs}}), \tag{6.29}$$

$$\sum_{k=1}^{M} y_{i,k}^{\text{obs}} \geq 1, \quad \forall i. \tag{6.30}$$

This ensures that at least one half-plane condition is robustly satisfied for each segment, following the approach used in the deterministic problem, but now taking uncertainty into account.

In fact, this endpoint-based verification can be interpreted geometrically as forming a "truncated cone" of safe configurations between X_i and X_{i+1}. By ensuring each endpoint, along with its associated uncertainty margin, lies within a safe half-plane, we create a robust corridor that envelops every feasible realization of the path segment. This safety guarantee holds because the uncertainty ellipse grows linearly between the endpoints, allowing the truncated cone to completely contain all possible trajectories between the verified endpoints. This geometric interpretation confirms that no realization of the robot's trajectory—under the given uncertainty bounds—deviates into unsafe regions.

Integrated Robust TAMP MILP

Combining all components—continuous-time parameterization, robust dynamics, scanning decisions, and robust obstacle avoidance—yields the robust TAMP MILP:

$$
\begin{aligned}
\min_{\bar{X}_i, \Delta X_i, \Delta T_i, c_i, y_{i,k}^{\text{obs}}, \epsilon_i} \quad & \sum_{i=1}^{N} \Delta T_i && (6.31) \\
\text{s.t.} \quad & X_{i+1} = X_i + \Delta X_i, && i = 1, \ldots, N-1, \\
& \Delta T_i = \Delta T_i^{\text{Move}} + c_i \Delta T_{\text{Scan}}, && \forall i, \\
& \|\Delta X_i\| \leq (1 - c_i) v_{\max} \Delta T_i^{\text{Move}}, && \forall i, \\
& a_k^{\text{obs}^T} \bar{X}_i - \epsilon_i \|a_k^{\text{obs}}\| \geq b_k^{\text{obs}} - M_{\text{big}}(1 - y_{i,k}^{\text{obs}}), && \forall i, k, \\
& a_k^{\text{obs}^T} \bar{X}_{i+1} - \epsilon_{i+1} \|a_k^{\text{obs}}\| \geq b_k^{\text{obs}} - M_{\text{big}}(1 - y_{i,k}^{\text{obs}}), && \forall i, k, \\
& \sum_{k=1}^{M} y_{i,k}^{\text{obs}} \geq 1, && \forall i, \\
& \epsilon_{i+1} = \epsilon_i + z\sigma_w \Delta T_i^{\text{Move}} - c_i \epsilon_i, && i = 1, \ldots, N-1, \\
& \sum_{i=1}^{N} \Delta T_i \leq T_{\max}, \\
& X_0 = X_0^*, \ X_N = X_N^*, \ \epsilon_0 = \epsilon_0^*, \\
& c_i \in \{0, 1\}, \ y_{i,k}^{\text{obs}} \in \{0, 1\}, && \forall i, k.
\end{aligned}
$$

Here: - $\bar{X}_i, \Delta X_i, \Delta T_i, \epsilon_i$ are continuous decision variables. - c_i and $y_{i,k}^{\text{obs}}$ are binary decision variables. - $v_{\max}, \Delta T_{\text{Scan}}, T_{\max}, z\sigma_w$, and M_{big} are known constants. - $X_0^*, X_N^*, \epsilon_0^*$ are given initial and terminal conditions.

This robust MILP ensures a solution that maintains feasibility under worst-case uncertainty realizations, leveraging scanning actions to control uncertainty growth. By using a continuous-time framework, we avoid the complexity of time discretization and produce a tractable solution that guarantees safety and enables effective long-range navigation under uncertainty.

6.5 Experiments

This section evaluates the proposed continuous-time robust TAMP, formulated as a convex MILP, through simulations and compares its performance with other similar approaches.

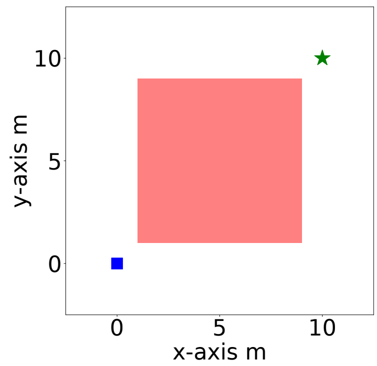


Fig. 6.4: Standard map

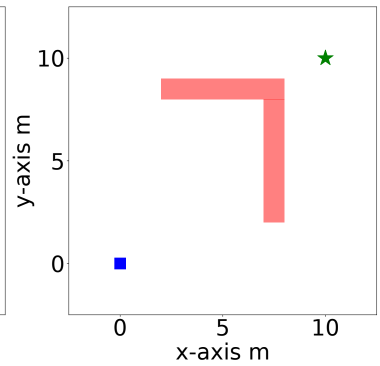


Fig. 6.5: Entrapped map

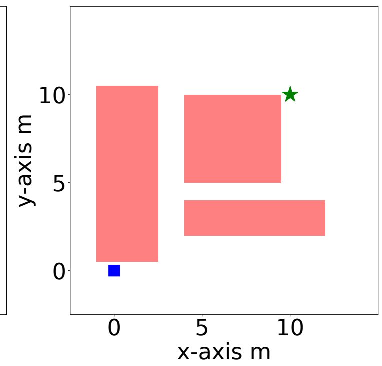


Fig. 6.6: Narrow map

Fig. 6.7: Obstacle scenarios (red) for simulation studies, with start and goal positions, colored in blue and green, respectively.

6.5.1 Metrics of Interest

We evaluate planners using three metrics: *Success Rate* (Succ.), *Execution Time* (ET), *Computation Time* (CT), and *Scan Count* (SC). *Success Rate* (%) measures the reliability of a planner by the proportion of trials in which the agent successfully reaches the specified goal. *Execution Time* (ET) [s] measures navigation efficiency as the total time required to move from the start position to the goal. In accordance with the capabilities of the EELS, the time costs for the $Move$ and *Scan* actions are set to 0.5 s and 100 s respectively. *Computation Time* (CT) [s] evaluates computational efficiency by summing the total time spent on planning. *Scan Count* (SC) directly measures how often the robot needs to perform scanning actions during navigation. While ET includes both movement and scanning time, SC specifically captures the frequency of localization updates required by each planner. Note that ET, CT, and SC are only calculated for trials that are counted as successful. All metrics are aggregated over multiple trials to ensure a statistically robust evaluation.

6.5.2 Experimental Setups

Our approach is compared with three planners that tackle the problem with different solution strategies:

Two-stage Planner: This approach follows the classic decoupling strategy of path planning and task scheduling, treating them separately. A* search is used to plan paths on a 2D gridmap with inflated obstacles, and scan scheduling is then performed.

Uncertainty is precomputed at each step, and scans are scheduled only when risk constraints are violated. After each scan, the planner repeats the path planning and scheduling processes.

MCTS-DPW Planner: MCTS-DPW [150] is a state-of-the-art POMDP solver that attempts to exactly solve the HPOMDP problem. Unlike our approach, it requires discretizing the action space for effective performance. In this setup, the control input is discretized into eight equal segments over a continuous state space. However, this baseline tends to exhibit myopic behavior, often failing to reach the goal due to the POMDP's limited planning horizon. To ensure a fair comparison, we incorporate techniques from [151], which extend planning horizons by building belief space roadmaps from the goal to the start.

Discrete-time MILP Planner: As detailed in [146], the discrete-time MILP planner aims to solve chance-constrained HPOMDP problems through convex MILP formulations that rely on belief MDP approximations and quadratic covariance growth models. This approach improves solution quality by simultaneously optimizing discrete actions and continuous control inputs. In addition, this formulation improves computational efficiency through simplification and linearization of the computationally intractable CC-HPOMDP problem. Our previous study confirmed that the MILP-based TAMP formulation outperforms both two-stage and MCTS-DPW planners in terms of solution quality, and achieves a lower computational burden than the MCTS-DPW planner.

We evaluate planners using a simulator that generates robot behavior following the dynamics described in Section 6.4. The two-stage and POMDP planners are implemented in Julia, using the Julia POMDP library [152]. The discrete- and continuous-time MILP planners are implemented in C++ using Google's or-tools [153] and Gurobi [154]. We integrate these multi-language configurations using ROS that connect planning and execution components through a client-service architecture. Note that all planners run on the same machine (AMD EPYC 7313 16-Core Processor, 251 GB of RAM, Ubuntu 20.04.6 LTS) to ensure a fair comparison of computation time.

We prepared three obstacle scenarios for testing, as shown in Fig. 6.7: 1) Standard (Std), 2) Entrapped (Ent), and 3) Narrow (Nar). Std-map assesses basic collision avoidance in open spaces. Ent-map tests planners against local minima entrapment due to finite horizon decision-making under uncertainty. Nar-map examines the trade-off between longer paths with fewer scans and shorter paths requiring more scans. In all scenarios, planners search from (0 m, 0 m) to (10 m, 10 m).

Table 6.2: Quantitative performance results of different planners across three obstacle scenarios

	Std-map			
Planners	ET↓	CT↓	SC↓	Succ.↑
Two-stage	281.22 ± 60.41	2.85 ± 6.02	2.69 ± 0.6	64
MCTS-DPW	430.93 ± 159.22	855.63 ± 174.65	4.07 ± 1.6	60
Discrete-time MILP	**125.96** ± 33.41	2.15 ± 0.48	**1.15** ± 0.4	**100**
Ours	158.52 ± 65.72	**0.06** ± 0.02	1.40 ± 0.7	**100**
	Ent-map			
Planners	ET↓	CT↓	SC↓	Succ.↑
Two-stage	166.86 ± 60.08	3.42 ± 6.39	1.55 ± 0.6	88
MCTS-DPW	349.47 ± 154.61	935.67 ± 177.72	3.29 ± 1.5	68
Discrete-time MILP	121.42 ± 27.94	3.63 ± 0.69	1.08 ± 0.3	**100**
Ours	**13.24** ± 0.50	**0.07** ± 0.01	**0.00** ± 0.0	**100**
	Nar-map			
Planners	ET↓	CT↓	SC↓	Succ.↑
Two-stage	412.53 ± 0.30	2.07 ± 5.28	4.00 ± 0.0	60
MCTS-DPW	381.50 ± 174.29	729.11 ± 170.73	3.53 ± 1.7	68
Discrete-time MILP	215.45 ± 0.55	22.18 ± 1.58	2.00 ± 0.0	84
Ours	**116.67** ± 0.35	**0.08** ± 0.01	**1.00** ± 0.0	**96**

ET = Execution Time (s), CT = Computation Time (s), SC = Scan Count, Succ. = Success Rate. All values are reported as mean ± standard deviation, computed over 25 successful runs. For risk-aware planners, we use a risk tolerance $\Delta = 0.1$. A z-value of 1.645 (corresponding to a 90% confidence interval) is employed for our robust planner.

6.5.3 Results and Discussions

Quantitative Comparison

Table 6.2 presents performance metrics for each planner over 25 Monte Carlo trials in three different obstacle configurations. The risk-aware planners use $\Delta = 0.1$, and the robust planner applies $z = 1.645$. These parameter values provide consistent comparisons of risk awareness and robustness.

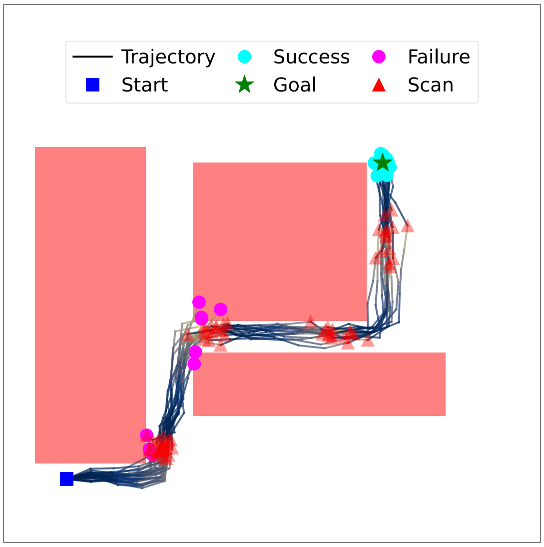
The proposed continuous-time MILP planner achieves the highest success rate among all methods and provides substantially shorter computation times. This balance between safety and computational efficiency suggests applicability to challenging navigation problems under state transition uncertainty. The joint solution of robust TAMP enables the generation of collision-free trajectories that ensure safety with minimal scan counts, which leads to reduced execution times across multiple scenarios. Although it does not always produce the shortest execution time, its performance remains competitive and often near-optimal.

The discrete-time MILP planner achieves a high success rate with relatively small computation times for simple scenarios, but this computational efficiency diminishes as scenario complexity grows. This occurs because adding temporal discretization expands the number of decision variables and constraints in an NP-hard MILP problem, which complicates optimization. The MCTS-DPW planner performs poorly in terms of efficiency. It suffers from high computational costs due to the complexity of exactly solving the hybrid POMDP under chance constraints. Such methods must deal with the combinatorial explosion of the hyperbelief space, making the problem intractable for more complex settings. The two-stage planner has relatively low computation times, but decoupling task and motion planning leads to less efficient and riskier navigation. This lower-quality solution is reflected in longer execution times and lower success rates, which highlight the importance of integrated approaches that address task and motion planning problems simultaneously.

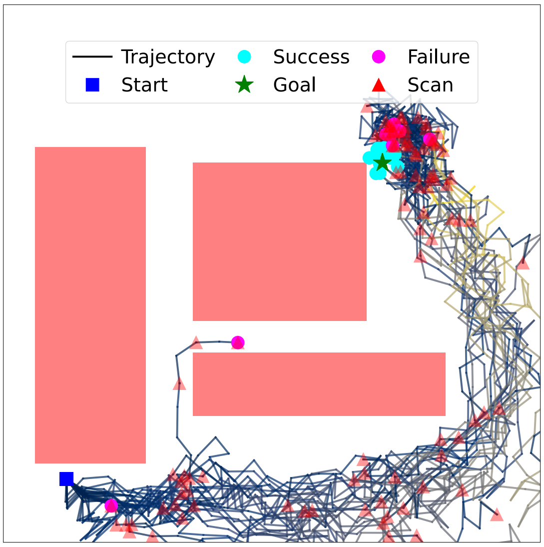
In summary, our continuous-time robust MILP planner for BLISS-TAMP simultaneously achieves high success rates, maintains competitive execution times, and obtains the lowest computation times of all the comparable planners. This result demonstrates the effectiveness of simultaneously addressing task and motion planning problems within a continuous-time planning formulation.

Qualitative Comparison

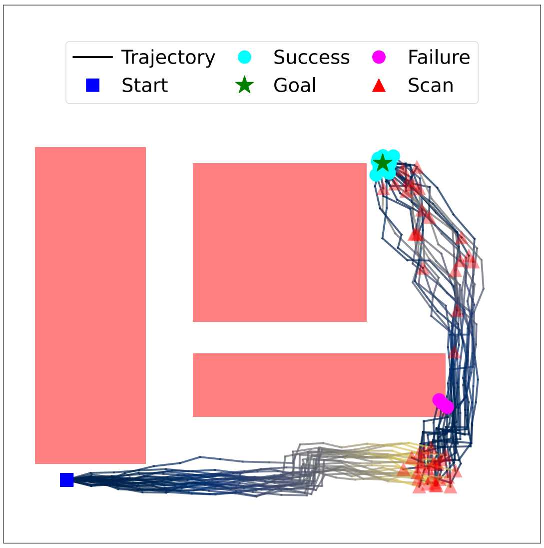
Fig. 6.8 presents Monte Carlo rollout results for the Nar map scenario, with trajectories color-coded according to the level of uncertainty growth during planning. The two-stage planner frequently fails in narrow passages due to its decoupled approach. It first identifies the shortest paths and only then schedules exteroceptive perception. Because the motion planning component prioritizes time efficiency, it generates trajectories that are overly risky and therefore lead to more hazardous navigation in terms of collision avoidance. In contrast, the MCTS-DPW planner produces more stochas-



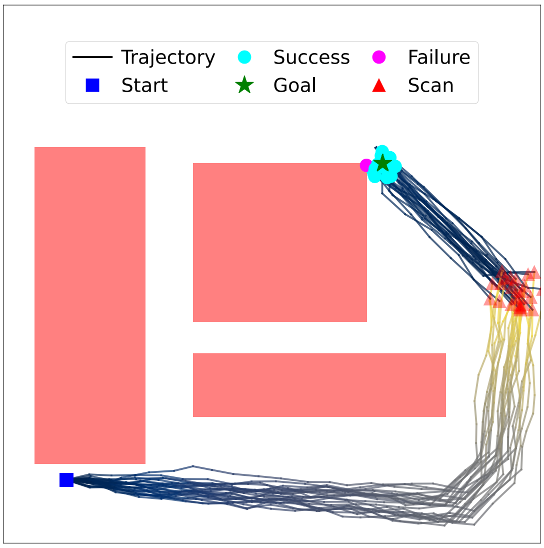
(a) Two-stage



(b) MCTS-DPW



(c) Discrete-time MILP



(d) **Ours**: Continuous-time MILP

Fig. 6.8: Comparison of planning and execution in a narrow map using four different planners. Trajectories are colored according to covariance growth during planning. Blue squares indicate start positions, green stars indicate goal positions, cyan circles indicate successful navigations, magenta circles indicate failures, and red triangles denote scanning locations.

tic execution compared to other methods, primarily because the anytime planning does not fully converge and it employs uncertainty sampling to mitigate the curse of history in hyperbelief spaces. Our continuous-time MILP planner supports safe and efficient navigation by maintaining sufficient clearance to avoid collisions during blind locomotion, while periodically scheduling scans to reduce uncertainty when it becomes

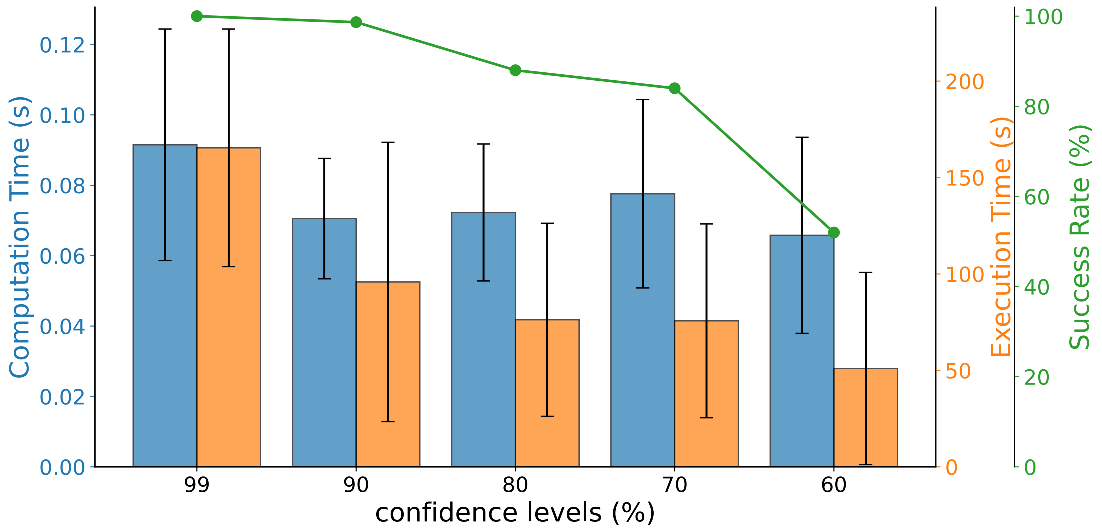


Fig. 6.9: Parameter study results for our continuous-time robust MILP among three different obstacle scenarios with varying confidence levels.

significant. This trend is similarly observed in the discrete-time MILP planner, which formulates TAMP with a similar strategy that jointly addresses scanning tasks and motion planning. However, while the discrete-time MILP planner adopts a risk-aware approach that incorporates uncertainty, it also tolerates certain levels of failure. As a result, it produces trajectories that are more prone to collisions.

Parameter Study

Fig. 6.9 demonstrates how our MILP planner's performance varies across three obstacle scenarios when using confidence levels ranging from 99% to 60%. Each confidence level corresponds to a specific z-value in $z\sigma_w$, chosen according to the quantile of a normal distribution. Higher confidence levels associated with larger z-values produce more conservative estimates of uncertainty growth. This gives higher success rates but longer execution times, as the planner tends to produce longer paths and perform more frequent scans to ensure substantial clearance against obstacles. In contrast, lower confidence levels with smaller z-values lead to more aggressive maneuvers. While this produces shorter execution times, it also leads to lower success rates due to reduced safety margins. Additionally, z influences computation time, as lower confidence levels expand the feasible solution space.

Limitations, Open Issues, and Possible Extensions

A key limitation of our planner is that its conservative behavior is controlled by user-defined confidence levels, as described in Section 6.5.3. There exists an inherent trade-off between execution time efficiency and safety guarantees; higher confidence levels lead to safer but slower navigation due to larger obstacle clearance and more frequent scanning actions, while lower levels result in faster but riskier trajectories. Our approach lacks a systematic method to adjust the optimal confidence level for a given scenario. This highlights the need for adaptive uncertainty awareness that optimizes conservative behavior based on local environmental conditions and task context.

While we have implemented our planner with lower-level autonomy components of the EELS platform, comprehensive quantitative evaluation of this integration in real-world settings remains an open challenge. Such evaluation would reveal how the complexity of the environment and the robotic system with real-world uncertainty affect the performance gap between simulation and reality.

A promising extension would be to explore risk-aware approaches to TAMP as an alternative to our robust formulation. Addressing the nonlinearity of chance constraints could enable different treatments of collision risk compared to the bounded uncertainty sets in our robust formulation, which potentially offers more resilient behaviors under state transition uncertainty.

6.6 Summary

We introduced BLISS-TAMP, a navigation strategy that leverages blind motion and intermittent scanning to improve robustness against localization failures in snake robots. By formulating the problem in continuous time rather than relying on discrete time steps, we significantly reduced computational complexity and accelerated planning speed. While our implementation focused on snake robot navigation, the continuous-time TAMP framework can benefit other robotic applications that require optimal scheduling of discrete tasks during continuous motion. For example, it can efficiently plan when and where to perform battery recharging tasks during rough terrain exploration or optimize the timing of resource-intensive sensing operations along a planned trajectory. Simulation results indicated that this approach outperformed existing planners not only in solution quality for time-efficient navigation and in shorter computation times to solve TAMP but also in safety due to its robust formulation for uncertain state transitions. This approach has already been integrated into the snake-type ice-world exploration robot EELS, and future work will quantitatively evaluate its performance under real-world conditions.

Chapter 7

Conclusions and Future Directions

7.1 Executive Summary

In this dissertation, we have studied resilient autonomy focusing on perception and planning algorithms for mobile robots operating in unstructured, rough terrain environments. Such environments introduce inherent perceptual uncertainty due to complex robot-terrain interactions, affecting both scene understanding for planning and state transitions during execution. We thus proposed a set of perception and planning methods that collectively address these uncertainties: perception describes uncertainty while planning exploits this uncertainty for resilient robotic navigation.

For uncertainty in scene understanding, we demonstrated the effectiveness of uncertainty quantification in perception through deterministic and probabilistic ML to describe known unknowns for informed decision-making. On the planning side, we showed how to exploit quantified uncertainty: 1) robustness, by incorporating uncertainty into traversability evaluation to prevent rover immobilization in deformable terrain, and 2) adaptability, by reducing uncertainty for more reliable long-term navigation. We also revealed that an end-to-end probabilistic ML model can unify uncertainty quantification, exploitation, and model adaptation within a single navigational framework.

For uncertainty in state transitions, we emphasized that resilience to localization failures is crucial for single-shot planetary exploration without prior knowledge or human assistance. The proposed BLISS system ensures resilient navigation by achieving both robustness against obstacle collisions and adaptability to unknown states through accurate localization. This is accomplished by solving an integrated TAMP formulation that ensures robust motion and incorporates periodic scans for localization.

Our findings demonstrated that uncertainty quantification enables both robust and adaptive navigation, and in turn, achieves resilient autonomy where robots balance safety and efficiency while exploring previously uncharted rough terrain. The presented perception and planning algorithms thus contribute to field robotics, where open-world scenarios inherently involve uncertainty in robotic perception.

7.2 Work Summary

The key techniques proposed in this dissertation are summarized as follows:

Chapter 2: Deterministic Uncertainty Handling for Traversability Prediction

This chapter presented a slip risk prediction method that accounts for both visual and geometric information by combining distinct ML models with deterministic uncertainty handling. Our method employs terrain-dependent slip regression models chosen by a terrain classifier, using slope angle information as input. The slip risk prediction focuses on uncertainties represented by discrepancies between actual and estimated slope angles, which determine the sharpness of the transition to rover immobilization. Through real-world data collection, we demonstrated that the proposed traversability evaluation method can incorporate uncertainty in geometric measurements to enable conservative assessment.

Chapter 3: Active Domain Adaptation via Robotic Information Gathering

This chapter presented *active traversability learning*, a new framework of learning traversability by planning risk-taking paths towards hazardous yet informative traverse data. The framework iteratively alternates between two stages: path planning to maximize information gain, and Gaussian process updates from in-situ measurements. We formulated risk-aware informative path planning by incorporating chance constraints into conventional informative path planning and developed a sampling-based path planner that solves the problem while accounting for robot immobilization risks. Through extensive simulation experiments, we demonstrated that the proposed framework enables robots to collect hazardous traverse data while avoiding immobilization.

Chapter 4: Probabilistic Traversability Fusion for Risk-aware Path Planning

This chapter presented a novel path planning algorithm that explicitly incorporates uncertainty in ML-based traversability prediction for heterogeneous terrain. We introduced *probabilistic fusion* of distinct ML models into a single probabilistic distribution represented as a mixture of Gaussian processes. The resulting multimodal distribution is then evaluated using statistical risk assessment methods, such as conditional value at risk, to enable conservative path planning. Through extensive simulation experiments using synthetic datasets that replicate the challenges in traversability prediction, we demonstrated that our proposed framework generates safer paths while maintaining efficiency in path execution time compared to the method proposed by Cunningham *et al.* [1].

Chapter 5: Robust and Adaptive Uncertainty Handling for Scene Understanding

This chapter presented a unified learning and planning framework for navigating rovers, with a focus on uncertainty handling of ML-based traversability prediction. We proposed *deep probabilistic traversability*, an end-to-end probabilistic ML model for traversability prediction. Through its structure of learning slip distributions directly from experience, the framework enables both uncertainty exploitation for conservative path planning and model adaptation during test-time navigation. Through extensive simulation experiments using synthetic datasets that focus on uncertainty from out-of-domain environmental features, we demonstrated that our approach enables safer rover traverse compared to the methods proposed by Cunningham *et al.* [1] and Endo *et al.* [2], particularly in terrains with ambiguous geometries and appearances.

Chapter 6: Integrated Task and Motion Planning for State Transition Uncertainty

This chapter presented a robust TAMP framework under uncertainty, aimed at achieving resilient navigation despite localization failures in robots with limited capabilities for simultaneous motion and precise localization. To enable blind motion with intermittently scheduled scans, we introduced a novel *continuous-time* planning formulation for a convex MILP that addresses computational intractability associated with TAMP problems under uncertainty. This formulation accelerates computation because it does not require explicit time discretization and thus reduces the number of optimization variables. Through simulation experiments under obstacle-rich scenarios, we demonstrated that our method derives safer and more time-efficient solutions with significantly less computation time compared to previous approaches.

7.3 Future Directions

Several directions for future work are proposed to sophisticate the presented perception and planning algorithms. The following highlights key future research directions.

Improvements of Uncertainty Quantification Models

One potential research direction is to advance uncertainty quantification in robotic perception using state-of-the-art techniques. For scene understanding, this involves incorporating advanced ML models to improve prediction performance and employing probabilistic modeling techniques for better uncertainty estimates. A promising alternative to CNN backbones is the transformer architecture [155], which leverages attention mechanisms to focus on the relevant input features. Several studies, such as Frey *et al.* [101] have applied transformers to predict traversability through semantic scene understanding, though they require addressing computational costs due to its large network architecture. Further, the latest probabilistic networks, such as evidential deep learning [156] and prior networks [157], can provide well-calibrated uncertainty estimates. For state transitions, it is crucial to consider more flexible, non-uniform distributions to properly capture uncertainty beyond standard Gaussian assumptions. Several studies have addressed this challenge by employing sampling-based methods [158] or moment-based representations [159] to model multimodal probability distributions.

Active Domain Adaptation during Navigation

The navigation framework presented in Chapter 5 could be extended by integrating the risk-taking behavior introduced in Chapter 3 to achieve more effective domain adaptation. This direction is promising for tighter integration of uncertainty exploitation and model adaptation described in this dissertation to simultaneously address robustness against inherent uncertainty and adaptiveness to unexplored regions. Deep probabilistic traversability is suitable for this objective, as it decomposes uncertainty into aleatoric and epistemic components. The epistemic component identifies areas that need additional training samples and thus enables more effective domain adaptation.

Integration with Lower-level Control Techniques

While this dissertation focuses on path and motion planning algorithms for long-range navigation using point mass approximation, further integrating these methods with lower-level controllers that account for robot dynamics and constraints (*e.g.*, torque limitations and actuator saturation) would enable more comprehensive robotic autonomy in real-world operations. Model Predictive Path Integral Control (MPPI) is an attractive control technique that can handle complex objectives through trajectory sampling without requiring differentiation. Several studies [85, 102, 103, 113, 114] have applied MPPI to off-road navigation tasks without using global path planning; however, these approaches still rely on long-range guidance due to the inherently myopic nature of MPPI solutions. To explore better integration of planning algorithms across different layers, we have developed BenchNav, a simulation platform for off-road navigation with probabilistic traversability modeling, as presented in Appendix A.

Hardware Verification for Sim-to-Real Transfer

A promising research direction is to further validate the effectiveness of the perception and planning algorithms presented in this dissertation by testing them on physical robot testbeds. In the context of scene understanding, this direction introduces additional challenges related to the efficiency of the proposed algorithms, given the limited onboard computational resources. From a perception standpoint, mobile robots in the field typically have fewer GPU resources than laboratory workstations, which complicates real-time adaptation. Nonetheless, these challenges may be mitigated through careful scheduling of knowledge updates and efficient resource allocation, for example by formulating a TAMP framework similar to the one presented in Chapter 6.

Moreover, real-world deployment must account for various uncertainties, such as diverse sensor noise characteristics, complex terrain deformation processes, and environmental factors affecting robot-terrain interaction, which could impact the performance metrics obtained in simulation. While this work focused on challenges in navigation on rough terrain by adopting simplified robot and environment models, it does not fully capture all these uncertainties in the simulation environment. To bridge the simulation-reality gap, systematic analysis of failure cases is crucial as the frequency and impact of different failure modes may vary in real environments. Understanding this gap through careful failure analysis and real-world validation would help identify critical scenarios for simulation testing and algorithmic improvements.

Operational Risk Management and Adaptation

Another important direction is to develop a framework for automated risk management that considers operational constraints and mission successes. While this dissertation focused on algorithmic approaches to risk assessment and mitigation, practical deployment requires careful consideration of how to adjust risk tolerance levels during actual operations. For example, space exploration missions typically begin with conservative strategies to ensure minimum success criteria, then gradually increase risk tolerance as operational experience accumulates.

Future work will focus on developing more sophisticated approaches to risk adaptation. In addition to automated adjustment of risk thresholds based on operational data, meta-level causal reasoning could enable more intelligent decision-making. For instance, hierarchical causal inference techniques could help identify complex patterns in failure modes, distinguishing between immediate environmental factors and higher-level systemic causes. Understanding these causal relationships would allow for more nuanced risk assessment and better-informed decisions about when and how to adjust risk tolerance. This deeper analytical approach would be particularly valuable for long-term autonomous missions where complex failure patterns may emerge over time.

Bibliography

[1] C. Cunningham, M. Ono, I. Nesnas, J. Yen, and W. L. Whittaker, "Locally-adaptive slip prediction for planetary rovers using gaussian processes," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 5487–5494, 2017.

[2] M. Endo, T. Taniai, R. Yonetani, and G. Ishigami, "Risk-aware path planning via probabilistic fusion of traversability prediction for planetary rovers on heterogeneous terrains," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 11 852–11 858, 2023.

[3] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Machine learning*, vol. 110, no. 3, pp. 457–506, 2021.

[4] X. Gao, J. Li, L. Fan, Q. Zhou, K. Yin, J. Wang, C. Song, L. Huang, and Z. Wang, "Review of wheeled mobile robots' navigation problems and application prospects in agriculture," *IEEE Access*, vol. 6, pp. 49 248–49 268, 2018.

[5] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[6] M. Tranzatto, T. Miki, M. Dharmadhikari, L. Bernreiter, M. Kulkarni, F. Mascarich, O. Andersson, S. Khattak, M. Hutter, R. Siegwart, and K. Alexis, "Cerberus in the darpa subterranean challenge," *Science Robotics*, vol. 7, no. 66, p. eabp9742, 2022.

[7] K. Ebadi, L. Bernreiter, H. Biggie, G. Catt, Y. Chang, A. Chatterjee, C. E. Denniston, S.-P. Deschênes, K. Harlow, S. Khattak, L. Nogueira, M. Palieri, P. Petráček, M. Petrlík, A. Reinke, V. Krátký, S. Zhao, A.-a. Agha-mohammadi, K. Alexis, C. Heckman, K. Khosoussi, N. Kottege, B. Morrell, M. Hutter, F. Pauling, F. Pomerleau, M. Saska, S. Scherer, R. Siegwart, J. L. Williams, and L. Carlone, "Present and future of slam in extreme environments: The darpa subt challenge," *IEEE Transactions on Robotics*, vol. 40, pp. 936–959, 2024.

[8] K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima, and S. Kawatsuma, "Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots," *Journal of Field Robotics*, vol. 30, no. 1, pp. 44–63, 2013.

[9] S. A. Chien, G. Visentin, and C. Basich, "Exploring beyond earth using space robotics," *Science Robotics*, vol. 9, no. 91, p. eadi6424, 2024.

[10] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "Anymal - a highly mobile and dynamic quadrupedal robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 38–44, 2016.

[11] A. Parness, N. Abcouwer, C. Fuller, N. Wiltsie, J. Nash, and B. Kennedy, "Lemur 3: A limbed climbing robot for extreme terrain mobility in space," in *IEEE International Conference on Robotics and Automation*, pp. 5467–5473, 2017.

[12] B. J. Hockman, A. Frick, R. G. Reid, I. A. Nesnas, and M. Pavone, "Design, control, and experimentation of internally-actuated rovers for the exploration of low-gravity planetary bodies," *Journal of Field Robotics*, vol. 34, no. 1, pp. 5–24, 2017.

[13] T. S. Vaquero, G. Daddi, R. Thakker, M. Paton, A. Jasour, M. P. Strub, R. M. Swan, R. Royce, M. Gildner, P. Tosi *et al.*, "Eels: Autonomous snake-like robot with task and motion planning capabilities for ice world exploration," *Science Robotics*, vol. 9, no. 88, p. eadh8332, 2024.

[14] A. Rankin, M. Maimone, J. Biesiadecki, N. Patel, D. Levine, and O. Toupet, "Driving curiosity: Mars rover mobility trends during the first seven years," in *2020 IEEE Aerospace Conference*, pp. 1–19, 2020.

[15] S. Goldberg, M. Maimone, and L. Matthies, "Stereo vision and rover navigation software for planetary exploration," in *Proceedings of the IEEE Aerospace Conference*, vol. 5, pp. 5–5, 2002.

[16] R. E. Arvidson, K. D. Iagnemma, M. Maimone, A. A. Fraeman, F. Zhou, M. C. Heverly, P. Bellutta, D. Rubin, N. T. Stein, J. P. Grotzinger, and A. R. Vasavada, "Mars science laboratory curiosity rover megaripple crossings up to sol 710 in gale crater," *Journal of Field Robotics*, vol. 34, no. 3, pp. 495–518, 2017.

[17] M. G. Bekker, *Introduction to Terrain-Vehicle Systems.* Ann Arbor, USA: University of Michigan Press, 1969.

[18] G. Ishigami, K. Nagatani, and K. Yoshida, "Slope traversal controls for planetary exploration rover on sandy terrain," *Journal of Field Robotics*, vol. 26, no. 3, pp. 264–286, 2009.

[19] J. Y. Wong, *Theory of ground vehicles.* John Wiley & Sons, 2008.

[20] I. Halatci, C. A. Brooks, and K. Iagnemma, "A study of visual and tactile terrain classification and classifier fusion for planetary exploration rovers," *Robotica*, vol. 26, pp. 767–779, 2008.

[21] C. A. Brooks and K. Iagnemma, "Self-supervised terrain classification for planetary surface exploration rovers," *Journal of Field Robotics*, vol. 29, no. 3, pp. 445–468, 2012.

[22] K. Otsu, M. Ono, T. J. Fuchs, I. Baldwin, and T. Kubota, "Autonomous terrain classification with co- and self-training approach," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 814–819, 2016.

[23] L.-P. Berczi, I. Posner, and T. D. Barfoot, "Learning to assess terrain from human demonstration using an introspective gaussian-process classifier," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3178–3185, 2015.

[24] F. Schilling, X. Chen, J. Folkesson, and P. Jensfelt, "Geometric and visual terrain classification for autonomous mobile navigation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2678–2684, 2017.

[25] S. Higa, Y. Iwashita, K. Otsu, M. Ono, O. Lamarre, A. Didier, and M. Hoffmann, "Vision-based estimation of driving energy for planetary rovers using deep learning and terramechanics," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3876–3883, 2019.

[26] A. Angelova, L. Matthies, D. Helmick, and P. Perona, "Learning and prediction of slip from visual information," *Journal of Field Robotics*, vol. 24, no. 3, pp. 205–231, 2007.

[27] K. Skonieczny, D. K. Shukla, M. Faragalli, M. Cole, and K. D. Iagnemma, "Data-driven mobility risk prediction for planetary rovers," *Journal of Field Robotics*, vol. 36, no. 2, pp. 475–491, 2019.

[28] M. Y. Yang and W. Förstner, "Plane detection in point cloud data," in *Proceedings of the International Conference on Machine Control Guidance*, pp. 95–104, 2010.

[29] S. Tanaka, K. Yamada, T. Ito, and T. Ohkawa, "Vehicle detection based on perspective transformation using rear-view camera," *International Journal of Vehicular Technology*, vol. 2011, 2011.

[30] D. Kim, S. M. Oh, and J. M. Rehg, "Traversability classification for ugv navigation: a comparison of patch and superpixel representations," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3166–3173, 2007.

[31] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2281, 2012.

[32] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[33] W.-Y. Loh, "Classification and regression trees," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 14–23, 2011.

[34] R. Gonzalez, M. Fiacchini, and K. Iagnemma, "Slippage prediction for off-road mobile robots via machine learning regression and proprioceptive sensing," *Robotics and Autonomous Systems*, vol. 105, pp. 85–93, 2018.

[35] R. Gonzalez, D. Apostolopoulos, and K. Iagnemma, "Slippage and immobilization detection for planetary exploration rovers via machine learning and proprioceptive sensing," *Journal of Field Robotics*, vol. 35, pp. 231–247, 2018.

[36] D. R. Freitag, A. J. Green, and K.-J. Melzer, "Performance evaluation of wheels for lunar vehicles," US Army Waterways Experiment Station, Vicksburg, MS, Technical Report M-70-2, 1970.

[37] C. M. Bishop, *Pattern Recognition and Machine Learning.* Berlin, Heidelberg: Springer-Verlag, 2006.

[38] G. Yamauchi, T. Noyori, K. Nagatani, and K. Yoshida, "Improvement of slope traversability for a multi-dof tracked vehicle with active reconfiguration of its joint forms," in *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics*, pp. 1–6, 2014.

[39] B. K. Muirhead, A. K. Nicholas, J. Umland, O. Sutherland, and S. Vijendran, "Mars sample return campaign concept status," *Acta Astronautica*, vol. 176, pp. 131–138, 2020.

[40] R. E. Arvidson, J. W. Ashley, J. F. Bell III, M. Chojnacki, J. Cohen, T. E. Economou, W. H. Farrand, R. Fergason, I. Fleischer, P. Geissler, R. Gellert, M. P. Golombek, J. P. Grotzinger, E. A. Guinness, R. M. Haberle, K. E. Herkenhoff, J. A. Herman, K. D. Iagnemma, B. L. Jolliff, J. R. Johnson, G. Klingelhöfer, A. H. Knoll, A. T. Knudson, R. Li, S. M. McLennan, D. W. Mittlefehldt, R. V. Morris, T. J. Parker, M. S. Rice, C. Schröder, L. A. Soderblom, S. W. Squyres, R. J. Sullivan, and M. J. Wolff, "Opportunity mars rover mission: Overview and selected results from purgatory ripple to traverses to endeavour crater," *Journal of Geophysical Research: Planets*, vol. 116, no. E7, 2011.

[41] M. G. Jadidi, J. V. Miro, and G. Dissanayake, "Sampling-based incremental information gathering with applications to robotic exploration and environmental monitoring," *International Journal of Robotics Research*, vol. 38, no. 6, pp. 658–685, 2019.

[42] D. Helmick, A. Angelova, and L. Matthies, "Terrain adaptive navigation for planetary rovers," *Journal of Field Robotics*, vol. 26, no. 4, pp. 391–410, 2009.

[43] M. Ono, T. J. Fuchs, A. Steffy, M. Maimone, and J. Yen, "Risk-aware planetary rover operation: Autonomous terrain classification and path planning," in *Proceedings of the IEEE Aerospace Conference*, pp. 1–10, 2015.

[44] M. Endo, S. Endo, K. Nagaoka, and K. Yoshida, "Terrain-dependent slip risk prediction for planetary exploration rovers," *Robotica*, vol. 39, no. 10, p. 1883–1896, 2021.

[45] G. Hedrick, N. Ohi, and Y. Gu, "Terrain-aware path planning and map update for mars sample return mission," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5181–5188, 2020.

[46] H. Inotsume and T. Kubota, "Adaptive terrain traversability prediction based on multi-source transfer gaussian processes," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2297–2304, 2021.

[47] N. R. J. Lawrance, J. J. Chung, and G. A. Hollinger, "Fast marching adaptive sampling," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 696–703, 2017.

[48] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies." *Journal of Machine Learning Research*, vol. 9, no. 2, 2008.

[49] J. Binney and G. S. Sukhatme, "Branch and bound for informative path planning," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2147–2154, 2012.

[50] J. Binney, A. Krause, and G. S. Sukhatme, "Optimizing waypoints for monitoring spatiotemporal phenomena," *International Journal of Robotics Research*, vol. 32, no. 8, pp. 873–888, 2013.

[51] S. M. LaValle and J. James J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[52] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[53] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *International Journal of Robotics Research*, vol. 33, no. 9, pp. 1271–1287, 2014.

[54] G. A. Hollinger, "Long-horizon robotic search and classification using sampling-based motion planning." in *Proceedings of Robotics: Science and Systems*, vol. 3, 2015.

[55] A. Viseras, D. Shutin, and L. Merino, "Online information gathering using sampling-based planners and gps: An information theoretic approach," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 123–130, 2017.

[56] A. Arora, P. M. Furlong, R. Fitch, S. Sukkarieh, and T. Fong, "Multi-modal active perception for information gathering in science missions," *Autonomous Robots*, vol. 43, no. 7, pp. 1827–1853, 2019.

[57] A. Candela, S. Kodgule, K. Edelson, S. Vijayarangan, D. R. Thompson, E. Noe Dobrea, and D. Wettergreen, “Planetary rover exploration combining remote and in situ measurements for active spectroscopic mapping,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 5986–5993, 2020.

[58] H. Inoue, M. Ono, S. Tamaki, and S. Adachi, “Active localization for planetary rovers,” in *Proceedings of the IEEE Aerospace Conference*, pp. 1–7, 2016.

[59] K. Otsu, A.-A. Agha-Mohammadi, and M. Paton, “Where to look? predictive perception with applications to planetary exploration,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 635–642, 2018.

[60] B. Luders, M. Kothari, and J. How, “Chance constrained rrt for probabilistic robustness to environmental uncertainty,” in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, p. 8160, 2010.

[61] A. Bry and N. Roy, “Rapidly-exploring random belief trees for motion planning under uncertainty,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 723–730, 2011.

[62] B. D. Luders, S. Karaman, and J. P. How, “Robust sampling-based motion planning with asymptotic optimality guarantees,” in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, p. 5097, 2013.

[63] S. U. Lee, R. Gonzalez, and K. Iagnemma, “Robust sampling-based motion planning for autonomous tracked vehicles in deformable high slip terrain,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2569–2574, 2016.

[64] H. Inotsume, T. Kubota, and D. Wettergreen, “Robust path planning for slope traversing under uncertainty in slip prediction,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3390–3397, 2020.

[65] M. Quann, L. Ojeda, W. Smith, D. Rizzo, M. Castanier, and K. Barton, “Chance constrained reachability in environments with spatially varying energy costs,” *Robotics and Autonomous Systems*, vol. 119, pp. 1–12, 2019.

[66] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning.* Cambridge, MA, USA: MIT press, 2006.

[67] J. D. Gammell and M. P. Strub, “Asymptotically optimal sampling-based motion planning methods,” *Annu. Rev. Control Robotics and Autonomous Systems*, vol. 4, pp. 295–318, 2021.

[68] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, “A survey of deep active learning,” *ACM Computing Surveys*, vol. 54, no. 9, 2021.

[69] J. A. Grant, M. P. Golombek, S. A. Wilson, K. A. Farley, K. H. Williford, and A. Chen, "The science process for selecting the landing site for the 2020 mars rover," *Planet. Space Sci.*, vol. 164, pp. 106–126, 2018.

[70] Y. Yokokohji, S. Chaen, and T. Yoshikawa, "Evaluation of traversability of wheeled mobile robots on uneven terrains by fractal terrain model," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2183–2188, 2004.

[71] M. H. Hecht, S. P. Kounaves, R. Quinn, S. J. West, S. M. Young, D. W. Ming, D. Catling, B. Clark, W. Boynton, J. Hoffman *et al.*, "Detection of perchlorate and the soluble chemistry of martian soil at the phoenix lander site," *Science*, vol. 325, no. 5936, pp. 64–67, 2009.

[72] B. Rothrock, R. Kennedy, C. Cunningham, J. Papon, M. Heverly, and M. Ono, "Spoc: Deep learning-based terrain classification for mars rover missions," in *Proceedings of AIAA SPACE*, p. 5539–5551, 2016.

[73] Y. Iwashita, K. Nakashima, J. Gatto, S. Higa, A. Stoica, N. Khoo, and R. Kurazume, "Virtual ir sensing for planetary rovers: Improved terrain classification and thermal inertia estimation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6302–6309, 2020.

[74] C. Cunningham, I. A. Nesnas, and W. L. Whittaker, "Improving slip prediction on mars using thermal inertia measurements," *Autonomous Robots*, vol. 43, no. 2, pp. 503–521, 2019.

[75] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[76] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, p. eaat8414, 2019.

[77] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020.

[78] A. Majumdar and M. Pavone, "How should a robot assess risk? towards an axiomatic theory of risk in robotics," in *Robotics Research*, pp. 75–84, 2020.

[79] A. Candela and D. Wettergreen, "An approach to science and risk-aware planetary rover exploration," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9691–9698, 2022.

[80] M. Endo and G. Ishigami, "Active traversability learning via risk-aware information gathering for planetary exploration rovers," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 855–11 862, 2022.

[81] A. Hakobyan, G. C. Kim, and I. Yang, "Risk-aware motion planning and control using cvar-constrained optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3924–3931, 2019.

[82] K. Ren, H. Ahn, and M. Kamgarpour, "Chance-constrained trajectory planning with multimodal environmental uncertainty," *IEEE Control Systems Letters*, vol. 7, pp. 13–18, 2023.

[83] V. D. Sharma, M. Toubeh, L. Zhou, and P. Tokekar, "Risk-aware planning and assignment for ground vehicles using uncertain perception from aerial vehicles," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 11 763–11 769, 2020.

[84] F. S. Barbosa, B. Lacerda, P. Duckworth, J. Tumova, and N. Hawes, "Risk-aware motion planning in partially known environments," in *Proceedings of IEEE Conference on Decision and Control*, pp. 5220–5226, 2021.

[85] X. Cai, M. Everett, J. Fink, and J. P. How, "Risk-aware off-road navigation via a learned speed distribution map," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2931–2937, 2022.

[86] A. S. McEwen, E. M. Eliason, J. W. Bergstrom, N. T. Bridges, C. J. Hansen, W. A. Delamere, J. A. Grant, V. C. Gulick, K. E. Herkenhoff, L. Keszthelyi *et al.*, "Mars reconnaissance orbiter's high resolution imaging science experiment (hirise)," *Journal of Geophysical Research: Planets*, vol. 112, no. E5, 2007.

[87] V. Tresp, "Mixtures of gaussian processes," in *Advances in Neural Information Processing Systems*, vol. 13, 2000.

[88] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, 2015.

[89] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[90] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of International Conference on Learning Representations*, 2015.

[91] M. Ono, B. Rothrock, E. Almeida, A. Ansar, R. Otero, A. Huertas, and M. Heverly, "Data-driven surface traversability analysis for mars 2020 landing site selection," in *Proceedings of IEEE Aerospace Conference*, pp. 1–12, 2016.

[92] V. Verma, M. W. Maimone, D. M. Gaines, R. Francis, T. A. Estlin, S. R. Kuhn, G. R. Rabideau, S. A. Chien, M. M. McHenry, E. J. Graser *et al.*, "Autonomous robotics is driving perseverance rover ' s progress on mars," *Science Robotics*, vol. 8, no. 80, p. eadi3099, 2023.

[93] S. W. Squyres, R. E. Arvidson, D. Bollen, J. F. Bell III, J. Brueckner, N. A. Cabrol, W. M. Calvin, M. H. Carr, P. R. Christensen, B. C. Clark *et al.*, "Overview of the opportunity mars exploration rover mission to meridiani planum: Eagle crater to purgatory ripple," *Journal of Geophysical Research: Planets*, vol. 111, no. E12, 2006.

[94] P. Filitchkin and K. Byl, "Feature-based terrain classification for littledog," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1387–1392, 2012.

[95] R. Schmid, D. Atha, F. Schöller, S. Dey, S. Fakoorian, K. Otsu, B. Ridge, M. Bjelonic, L. Wellhausen, M. Hutter *et al.*, "Self-supervised traversability prediction by learning to reconstruct safe terrain," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 12 419–12 425, 2022.

[96] J. Seo, S. Sim, and I. Shim, "Learning off-road terrain traversability with self-supervisions only," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4617–4624, 2023.

[97] S. Jung, J. Lee, X. Meng, B. Boots, and A. Lambert, "V-strong: Visual self-supervised traversability learning for off-road navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1766–1773, 2024.

[98] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, "Where should i walk? predicting terrain properties from images via self-supervised learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1509–1516, 2019.

[99] M. G. Castro, S. Triest, W. Wang, J. M. Gregory, F. Sanchez, J. G. Rogers, and S. Scherer, "How does it feel? self-supervised costmap learning for off-road vehicle traversability," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 931–938, 2023.

[100] E. Chen, C. Ho, M. Maulimov, C. Wang, and S. Scherer, "Learning-on-the-drive: Self-supervised adaptation of visual offroad traversability models," *arXiv:2306.15226*, 2023.

[101] J. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter, "Fast traversability estimation for wild visual navigation," in *Proceedings of Robotics: Science and Systems*, 2023.

[102] M. V. Gasparino, A. N. Sivakumar, Y. Liu, A. E. Velasquez, V. A. Higuti, J. Rogers, H. Tran, and G. Chowdhary, "Wayfast: Navigation with predictive traversability in the field," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 651–10 658, 2022.

[103] M. V. Gasparino, A. N. Sivakumar, and G. Chowdhary, “Wayfaster: a self-supervised traversability prediction for increased navigation awareness,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 8486–8492, 2024.

[104] X. Meng, N. Hatch, A. Lambert, A. Li, N. Wagener, M. Schmittle, J. Lee, W. Yuan, Z. Chen, S. Deng *et al.*, “Terrainnet: Visual modeling of complex terrain for high-speed, off-road navigation,” in *Proceedings of Robotics: Science and Systems*, 2023.

[105] D. J. MacKay, “A practical bayesian framework for backpropagation networks,” *Neural Comput.*, vol. 4, no. 3, pp. 448–472, 1992.

[106] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *Proc. Int. Conf. Mach. Learn.*, vol. 48, pp. 1050–1059, 2016.

[107] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[108] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher *et al.*, “A survey of uncertainty in deep neural networks,” *Artif. Intell. Rev.*, pp. 1–77, 2023.

[109] R. Takemura and G. Ishigami, “Uncertainty-aware trajectory planning: Using uncertainty quantification and propagation in traversability prediction of planetary rovers,” *IEEE Robotics and Automation Magazine*, pp. 2–12, 2023.

[110] H. Lee, J. Kwon, and C. Kwon, “Learning-based uncertainty-aware navigation in 3d off-road terrains,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 10 061–10 068, 2023.

[111] D. D. Fan, K. Otsu, Y. Kubo, A. Dixit, J. Burdick, and A.-A. Agha-Mohammadi, “Step: Stochastic traversability evaluation and planning for safe off-road navigation,” in *Proc. Robotics: Science and Systems*, 2021.

[112] W. Feng, L. Ding, R. Zhou, C. Xu, H. Yang, H. Gao, G. Liu, and Z. Deng, “Learning-based end-to-end navigation for planetary rovers considering non-geometric hazards,” *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 4084–4091, 2023.

[113] X. Cai, M. Everett, L. Sharma, P. R. Osteen, and J. P. How, “Probabilistic traversability model for risk-aware motion planning in off-road environments,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 11 297–11 304, 2023.

[114] X. Cai, S. Ancha, L. Sharma, P. R. Osteen, B. Bucher, S. Phillips, J. Wang, M. Everett, N. Roy, and J. P. How, “EVORA: Deep evidential traversability learning for risk-aware off-road autonomy,” *IEEE Transactions on Robotics*, pp. 1–20, 2024.

[115] T. Kim, J. Mun, J. Seo, B. Kim, and S. Hong, "Bridging active exploration and uncertainty-aware deployment using probabilistic ensemble neural network dynamics," in *Proceedings of Robotics: Science and Systems*, 2023.

[116] W. Zhang, L. Shen, W. Zhang, and C.-S. Foo, "Few-shot adaptation of pre-trained networks for domain shift," in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1665–1671, 2022.

[117] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, 1968.

[118] H. Kim and J.-S. Lee, "Scalable monotonic neural networks," in *Proceedings of the International Conference on Learning Representations*, 2024.

[119] W. Talbot, J. Nash, M. Paton, E. Ambrose, B. Metz, R. Thakker, R. Etheredge, M. Ono, and V. Ila, "Principled icp covariance modelling in perceptually degraded environments for the eels mission concept," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 10 763–10 770, 2023.

[120] F. Lagriffoul and B. Andres, "Combining task and motion planning: A culprit detection problem," *International Journal of Robotics Research*, vol. 35, no. 8, pp. 890–927, 2016.

[121] N. T. Dantam, Z. K. Kingston, S. Chaudhuri, and L. Kavraki, "Incremental task and motion planning: A constraint-based approach," in *Proceedings of Robotics: Science and Systems*, vol. 12, p. 00052, 2016.

[122] K. He, M. Lahijanian, L. E. Kavraki, and M. Y. Vardi, "Towards manipulation planning with temporal logic specifications," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 346–352, 2015.

[123] A. Akbari, J. Rosell *et al.*, "Task and motion planning using physics-based reasoning," in *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 1–7, 2015.

[124] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, "Combined task and motion planning through an extensible planner-independent interface layer," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 639–646, 2014.

[125] S. Lo, S. Zhang, and P. Stone, "The petlon algorithm to plan efficiently for task-level-optimal navigation," *Journal of Artificial Intelligence Research*, vol. 67, 2020.

[126] C. Rodríguez and R. Suárez, "Combining motion planning and task assignment for a dual-arm system," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4238–4243, 2016.

[127] E. Fernandez-Gonzalez, E. Karpas, and B. C. Williams, "Scottyactivity: Mixed discrete-continuous planning with convex optimization," *Journal of Artificial Intelligence Research*, vol. 62, pp. 579–664, 2018.

[128] M. Diab, A. Akbari, M. Ud Din, and J. Rosell, "Pmk—a knowledge processing framework for autonomous robotics perception and manipulation," *Sensors*, vol. 19, no. 5, p. 1166, 2019.

[129] S. Cambon, R. Alami, and F. Gravot, "A hybrid approach to intricate motion, manipulation and task planning," *International Journal of Robotics Research*, vol. 28, no. 1, pp. 104–126, 2009.

[130] C. R. Garrett, T. Lozano-P., and L. P. Kaelbling, "Ffrob: An efficient heuristic for task and motion planning," in *Proceedings of the International Workshop on the Algorithmic Foundations of Robotics*, pp. 179–195, 2015.

[131] A. Akbari, M. Gillani, and J. Rosell, "Reasoning-based evaluation of manipulation actions for efficient task planning," in *Proceedings of the Iberian Robotics Conference: Advances in Robotics*, pp. 69–80, 2015.

[132] A. Akbari, Muhayyuddin, and J. Rosell, "Task planning using physics-based heuristics on manipulation actions," in *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 1–8, 2016.

[133] A. Akbari, Muhayyuddin, and . Rosell, J, "Knowledge-oriented task and motion planning for multiple mobile robots," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 31, no. 1, pp. 137–162, 2019.

[134] A. Akbari, F. Lagriffoul, and J. Rosell, "Combined heuristic task and motion planning for bi-manual robots," *Autonomous Robots*, vol. 43, no. 6, pp. 1575–1590, 2019.

[135] A. Nouman, I. F. Yalciner, E. Erdem, and V. Patoglu, "Experimental evaluation of hybrid conditional planning for service robotics," in *Proceedings of the International Symposium on Experimental Robotics*, pp. 692–702, 2017.

[136] A. Akbari, M. Diab, and J. Rosell, "Contingent task and motion planning under uncertainty for human–robot interactions," *Applied Sciences*, vol. 10, no. 5, 2020.

[137] S. Mochizuki, Y. Kawasaki, and M. Takahashi, "Protamp: Probabilistic task and motion planning considering human action for harmonious collaboration," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2159–2166, 2022.

[138] P. Santana, T. Vaquero, C. L. McGhan, C. Toledo, E. Timmons, B. Williams, and R. Murray, "Risk-aware planning in hybrid domains: An application to autonomous planetary rovers," in *Proceedings of the AIAA SPACE*, 2016.

[139] X. Huang, A. Jasour, M. Deyo, A. Hofmann, and B. C. Williams, "Hybrid risk-aware conditional planning with applications in autonomous vehicles," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 3608–3614, 2018.

[140] B. Ayton, E. Timmons, R. Camilli, and B. C. Williams, "Risk aware adaptive sampling for the search for life in ocean worlds," in *Proceedings of the Astrobiology Science Conference*, 2019.

[141] B. Ayton, B. C. Williams, and R. Camilli, "Measurement maximizing adaptive sampling with risk bounding functions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 7511–7519, 2019.

[142] L. Janson, E. Schmerling, and M. Pavone, "Monte carlo motion planning for robot trajectory optimization under uncertainty," in *Robotics Research*, 2017, pp. 343–361.

[143] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE Transactions on Robotics*, vol. 26, no. 3, p. 502–517, jun 2010.

[144] L. Blackmore, M. Ono, and B. C. Williams, "Chance-constrained optimal path planning with obstacles," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1080–1094, 2011.

[145] L. Blackmore and M. Ono, "Convex chance constrained predictive control without sampling," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, p. 5876, 2009.

[146] A. Jasour, G. Daddi, M. Endo, T. S. Vaquero, M. Paton, M. P. Strub, S. Corpino, M. Ingham, M. Ono, and R. Thakker, "Risk-aware integrated task and motion planning for versatile snake robots under localization failures," in *Under Review for the IEEE International Conference on Robotics and Automation*, 2025.

[147] H. Ponte, M. Queenan, C. Gong, C. Mertz, M. Travers, F. Enner, M. Hebert, and H. Choset, "Visual sensing for developing autonomous behavior in snake robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2779–2784, 2014.

[148] R. Thakker, M. Paton, M. P. Strub, M. Swan, G. Daddi, R. Royce, P. Tosi, M. Gildner, T. Vaquero, M. Veismann *et al.*, "Eels: Towards autonomous mobility in extreme terrain with a versatile snake robot with resilience to exteroception failures," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 9886–9893, 2023.

[149] R. Thakker, M. Paton, B. Jones, G. Daddi, R. Royce, M. Swan, M. Strub, S. Aghli, H. Zade, Y. K. Nakka *et al.*, "To boldly go where no robots have gone before – part 4: Neo autonomy for robustly exploring unknown, extreme environments with versatile robots," in *Proceedings of the AIAA SCITECH Forum*, 2024.

[150] A. Couëtoux, J. Hoock, N. Sokolovska, O. Teytaud, and N. Bonnard, "Continuous upper confidence trees," in *Proceedings of the International Conference on Learning and Intelligent Optimization*, pp. 433–445, 2011.

[151] H. Kim, R. Thakker, and A. Agha-Mohammadi, "Bi-directional value learning for risk-aware planning under uncertainty," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2493–2500, 2019.

[152] M. Egorov, Z. N. Sunberg, E. Balaban, T. A. Wheeler, J. K. Gupta, and M. J. Kochenderfer, "POMDPs.jl: A framework for sequential decision making under uncertainty," *Journal of Machine Learning Research*, vol. 18, no. 26, pp. 1–5, 2017.

[153] L. Perron and V. Furnon, Google. https://developers.google.com/optimization/. , as of

[154] Gurobi Optimization, LLC, 2024. https://www.gurobi.com. , as of

[155] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the International Conference on Neural Information Processing Systems*, p. 6000–6010, 2017.

[156] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *Proceedings of the International Conference on Neural Information Processing Systems*, p. 3183–3193, 2018.

[157] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, p. 7047–7058, 2018.

[158] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 502–517, 2010.

[159] A. Wang, A. Jasour, and B. C. Williams, "Non-gaussian chance-constrained trajectory planning for autonomous vehicles under agent uncertainty," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6041–6048, 2020.

[160] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics and Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[161] V. Verma, M. W. Maimone, D. M. Gaines, R. Francis, T. A. Estlin, S. R. Kuhn, G. R. Rabideau, S. A. Chien, M. M. McHenry, E. J. Graser *et al.*, "Autonomous robotics is driving perseverance rover ' s progress on mars," *Science Robotics*, vol. 8, no. 80, p. eadi3099, 2023.

[162] G. Ishigami, K. Nagatani, and K. Yoshida, "Path planning for planetary exploration rovers and its evaluation based on wheel slip dynamics," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2361–2366, 2007.

[163] S. Daftry, N. Abcouwer, T. Del Sesto, S. Venkatraman, J. Song, L. Igel, A. Byon, U. Rosolia, Y. Yue, and M. Ono, "MLNav: Learning to safely navigate on martian terrains," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5461–5468, 2022.

[164] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105–1118, 2009.

[165] R. Takemura and G. Ishigami, "Traversability-based trajectory planning with quasi-dynamic vehicle model in loose soil," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 8411–8417, 2021.

[166] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic mpc for model-based reinforcement learning," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1714–1721, 2017.

[167] J. Gibson, B. Vlahov, D. Fan, P. Spieler, D. Pastor, A.-a. Agha-mohammadi, and E. A. Theodorou, "A multi-step dynamics modeling framework for autonomous driving in multiple environments," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 7959–7965, 2023.

[168] T. Han, A. Liu, A. Li, A. Spitzer, G. Shi, and B. Boots, "Model predictive control for aggressive driving over uneven terrain," *arXiv:2311.12284*, 2023.

[169] S. Triest, M. Sivaprakasam, S. J. Wang, W. Wang, A. M. Johnson, and S. Scherer, "TartanDrive: A large-scale dataset for learning off-road dynamics models," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2546–2552, 2022.

[170] S. Sharma, L. Dabbiru, T. Hannis, G. Mason, D. W. Carruth, M. Doude, C. Goodin, C. Hudson, S. Ozier, J. E. Ball *et al.*, "CaT: Cavs traversability dataset for off-road autonomous driving," *IEEE Access*, vol. 10, pp. 24 759–24 768, 2022.

[171] M. Sivaprakasam, S. Triest, M. G. Castro, M. Nye, M. Maulimov, C. Ho, P. Maheshwari, W. Wang, and S. Scherer, "TartanDrive 1.5: Improving large multimodal robotics dataset collection and distribution," in *Proceedings of the International Conference on Robotics and Automation Workshop Pretraining for Robotics*, 2023.

[172] M. Sivaprakasam, P. Maheshwari, M. G. Castro, S. Triest, M. Nye, S. Willits, A. Saba, W. Wang, and S. Scherer, "TartanDrive 2.0: More modalities and better infrastructure to further self-supervised learning research in off-road driving tasks," *arXiv:2402.01913*, 2024.

[173] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 32, 2019.

[174] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv:1606.01540*, 2016.

[175] D. D. Fan, A.-a. Agha-mohammadi, and E. A. Theodorou, "Learning risk-aware costmaps for traversability in challenging environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 279–286, 2022.

[176] S. Triest, M. G. Castro, P. Maheshwari, M. Sivaprakasam, W. Wang, and S. Scherer, "Learning risk-aware costmaps via inverse reinforcement learning for off-road navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 924–930, 2023.

[177] M. Ono, M. Heverly, B. Rothrock, E. Almeida, F. Calef, T. Soliman, N. Williams, H. Gengl, T. Ishimatsu, A. Nicholas *et al.*, "Mars 2020 site-specific mission performance analysis: Part 2. surface traversability," in *Proceedings of the AIAA SPACE*, 2018.

[178] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[179] M. G. Castro, S. Triest, W. Wang, J. M. Gregory, F. Sanchez, J. G. Rogers, and S. Scherer, "How does it feel? self-supervised costmap learning for off-road vehicle traversability," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 931–938, 2023.

[180] L. Gan, J. W. Grizzle, R. M. Eustice, and M. Ghaffari, "Energy-based legged robots terrain traversability modeling via deep inverse reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8807–8814, 2022.

[181] G. B. Margolis, X. Fu, Y. Ji, and P. Agrawal, "Learning to see physical properties with active sensing motor policies," *arXiv:2311.01405*, 2023.

Appendix A

BenchNav: Off-road Navigation Simulator

A.1 Introduction

Mobile robots unchain us from labor-intensive, hazardous activities *in the field*, ranging from environmental monitoring [160], search and rescue [8], to planetary exploration [161]. Autonomous navigation is critical for reliable decision-making, enabling safe and efficient mission operations without consistent human intervention. However, off-road navigation is fraught with risks in unstructured terrains, such as robots getting stuck in deformable terrain, tipping over on steep slopes, or colliding with both positive and negative obstacles. *Traversability* is thus an essential criterion to quantify the level of robot ability to traverse, ranging from safe to hazardous, in unstructured off-road environments.

Off-road navigation has evolved with path and motion planning algorithms that incorporate traversability either through theoretical or machine learning (ML)-based modeling. *Search-based* path planning, valid in discretized graph environments, is notable for its resolution optimality. It can generate global paths for navigation in extreme environments, such as planetary surfaces [2, 15, 161–163] and subterranean domains [111], making it suitable for real-world applications, including NASA's AutoNav [15] and Enav [161] systems. Nevertheless, search-based methods often miss robot dynamics, leading to challenges in translating global paths to local motions. *Sampling-based* motion planning has become another common approach, capable of handling high-dimensional spaces through random environmental sampling to con-

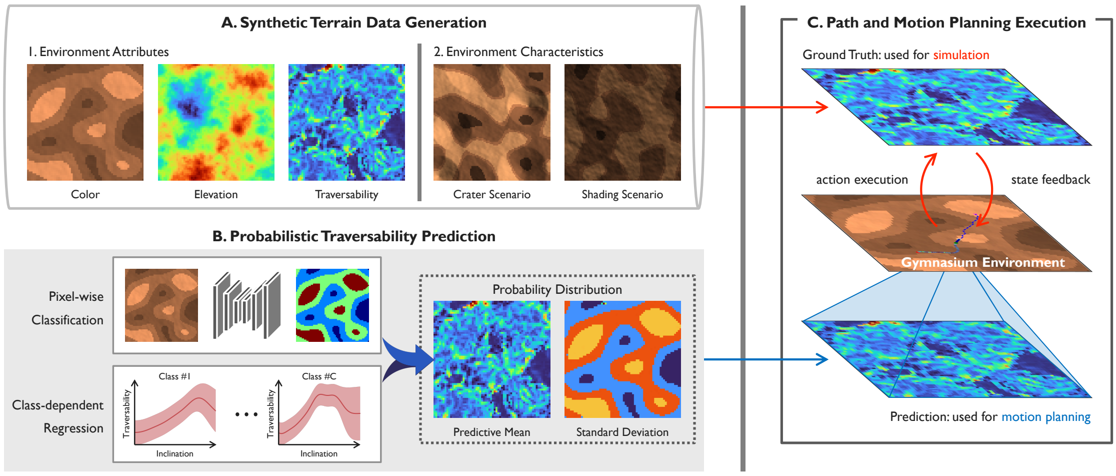


Fig. A.1: BenchNav, a platform designed to solve off-road navigation problems, consists of A) synthetic terrain data generation, B) probabilistic traversability prediction based on the classify-then-regress method, and C) path and motion planning execution with given problem formulations and solvers.

struct feasible robot states. Its flexible representation in configuration space allows us to apply complex dynamical models in motion planning, such as closed-loop rapidly-exploring random trees (CL-RRT) [164] and its variants in off-road navigation [109,165]. Still, they often indirectly include dynamical models and require substantial computational effort to reach asymptotic optimality. *Optimization-based* algorithms, in contrast, enable the direct integration of dynamical models into motion planning, resulting in optimal state and control sequences. Model predictive path integral control (MPPI) [166], a sampling-based optimal control solver, is favored for aggressive off-road navigation [85,102,103,113,114,167,168], offering rapid trajectory optimization and the ability to handle non-differentiable objectives. These techniques focus on generating sequential, short-horizon solutions while often failing to achieve global optimality.

We here argue that off-road autonomy requires a well-aligned integration of planning algorithms and traversability modeling for robot dynamics. Along with the advances in ML models and growing dataset availability [169–172], many studies push forward data-driven traversability prediction and its application to motion planning. However, the lack of standardized benchmarking for planning raises the question: *How can we select the suitable algorithm for designing an off-road navigation system, given their unique differences?* This is often empirically tailored to specific scenarios, though quantitative comparison across different domains is a tedious yet necessary process.

To address the above challenge, we present BenchNav, an open-source, PyTorch-based [173] simulation platform, designed for **Bench**marking off-road **Nav**igation algorithms. Leveraging Gymnasium [174], BenchNav equips three key features to tackle off-road navigation problems as follows:

- Synthetic data generation replicates natural terrain, embedding the challenges of traversability prediction,
- Built-in ML models for probabilistic traversability prediction from given environmental data,
- Path and motion planning algorithm executions unified with robot models factoring in uncertain traversability.

The synthesized process, encompassing data preparation, traversability modeling, and planning, ensures easy and consistent navigation simulations across various algorithms. We perform simulation experiments in deformable off-road environments, focusing on vehicle slip as a typical metric for traversability modeling. The use of different path and motion planning algorithms demonstrates BenchNav's versatility.

A.2 Problem Statement

We state the off-road navigation problem for mobile robots as one embracing traversability prediction and motion planning. Unstructured environments present complex physical interactions with robots, leading to deviations between commanded and actual velocities. Under significant disturbances, the discrete-time system is given as $\mathbf{s}_{t+1} = F\left(\mathbf{s}_t, \mathbf{a}_t, \boldsymbol{\lambda}\right)$, where t denotes discrete time steps. Here, $\mathbf{s}_t \in \mathbb{R}^n$ represents the state vector, $\mathbf{a}_t \in \mathbb{R}^m$ the action vector serving as the control input to the robot, and $\boldsymbol{\lambda} \in [0,1]^p$ the traversability coefficient vector, scales $\mathbf{a}_t$. The state transition function F integrates these inputs to update the state vector, determining the system's response accounting for terrain interactions.

The traversability prediction objective is to create a model $\mathcal{M}$, mapping environmental observations $\mathbf{o} \in O$ to traversability coefficients $\boldsymbol{\lambda} \in \Lambda$, where O and Λ are sets of all environmental observations and traversability coefficients, respectively. ML is a powerful tool for discovering latent models by learning general correlations from training data. Recent studies also reveal the value of quantifying uncertainty in traversability learning [80,175,176], arising from inherent observation noise and limited training data.

We thus adopt probabilistic ML models as the basis for traversability prediction used across planning algorithms.

Besides, the motion planning objective is to find a sequence of feasible actions $\mathbf{a}_t = \pi(\mathbf{s}_t, \hat{\boldsymbol{\lambda}})$ towards its destination, with a policy π that translates current states and traversability coefficients into the state transition dynamics. The motion planning problem is formulated to optimize user-specified objectives, such as time efficiency, subject to certain constraints. Despite the various solvers available, we emphasize the need to exploit uncertainty in traversability prediction. Following earlier studies [2, 85, 113, 114], we employ statistical methods for risk inference, such as conditional value at risk (CVaR) [78], to transform probability distributions into deterministic inputs. This approach bridges traversability prediction and motion planning, ensuring consistent execution in solving the off-road navigation problem.

A.3 BenchNav

This section details BenchNav, a simulation platform tailored for off-road navigation. We developed BenchNav on top of Gymnasium (formerly OpenAI Gym), leveraging its capabilities for sequential decision-making and scalable simulation of diverse environments. BenchNav's implementation in PyTorch further enables GPU acceleration, ensuring efficient computation for ML models and planning algorithms throughout the platform. The overall simulator pipeline, depicted in Fig. A.1, unfolds as follows: It first generates controlled datasets containing O and their respective Λ, engineered to catch the difficulties present in traversability prediction. Then, a set of all probability distributions for traversability coefficients, denoted as $\mathbb{P}(\hat{\boldsymbol{\lambda}}|\mathbf{o})$, is formed by initially categorizing symbolic terrain classes from appearance features, followed by predicting class-dependent latent traversability (LT) functions from geometric features. The final stage involves running planning algorithms to find $\mathbf{a}_t = \pi(\mathbf{s}_t, \hat{\boldsymbol{\lambda}})$, with given problem formulations and solvers that incorporate risk inference metrics for uncertain traversability.

A.3.1 Synthetic Terrain Data Generation

We synthesize a top-down 2.5D terrain map instance, replicating pixel-wise appearance and geometric features, alongside their latent traversability coefficients. In off-road scenarios, appearance features indicate surface properties, determining distinct traversability trends among terrain classes. Geometric features also dominate the variation in traversability within each class. The data is thereby paired: RGB color data with their corresponding terrain classes $c \in C$, where C is the set of all terrain classes, and elevation data coupled with class-dependent LT functions $f_c(\psi)$, where ψ is the inclination derived by the Horn method [177].

For each map instance, occupancy ratios dictate terrain class distribution; we create such a distribution with Perlin noise to assign color and terrain classes based on the clusters. We then simulate rough terrain elevation with fractal terrain modeling [70] to compute $\lambda = f_c(\psi) + \epsilon_c$, where ϵ represents additive zero-mean Gaussian noise, and to apply shading to the color data due to elevation changes. We consequently get a map of colors, elevations, and traversability coefficients, depicted in Fig. A.1A1. Environmental features are readily adjusted to simulate challenges in traversability prediction, such as ambiguous appearance from intense shading and irregular geometry in crater-like terrains, shown in Fig. A.1A2.

A.3.2 Probabilistic Traversability Prediction

BenchNav employs two types of pre-trained ML models: 1) a terrain classifier for predicting pixel-wise terrain classes and 2) Gaussian processes (GPs) [66] for estimating class-dependent LT functions with uncertainties. These distinctive ML models are integrated into a single probability distribution via mixtures of GPs [87], representing probabilistic traversability at $\mathbf{s}$ as follows:

$$\mathbb{P}_{\mathbf{s}}(\hat{\lambda}) = \sum_{c \in C} \mathbb{P}_{\mathbf{s}}\,(c)\,\mathbb{P}_c(\hat{\lambda} \mid \psi_{\mathbf{s}}), \tag{A.1}$$

where $\mathbb{P}_c(\hat{\lambda} \mid \psi_{\mathbf{s}})$ denotes class-dependent GPs, weighted by a categorical distribution $\mathbb{P}_{\mathbf{s}}(c)$ from the classifier. This probabilistic fusion [2] constructs $\mathbb{P}(\hat{\boldsymbol{\lambda}}|\mathbf{o})$, as shown in Fig. A.1B.

While any model capable of predicting pixel-wise categorical distributions is suitable for terrain classification, we opt for the U-Net model [88] with a ResNet-18 backbone [89]. Users also have the option of selecting a single GP that corresponds to the most likely class $c^* = \arg\max_c \mathbb{P}_{\mathbf{s}}(c)$ in eq. (A.1), rather than summing over multiple GPs.

A.3.3 Path and Motion Planning Execution

The last stage simulates off-road robot navigation from a start state $\mathbf{s}_{\text{start}}$ to a goal state $\mathbf{s}_{\text{goal}}$ within the allotted finite positive time budget T. The motion planning problem in off-road environments is given as follows:

$$\begin{aligned}
\text{Find:}\ & \{\mathbf{a}_t\}_{t=0}^{T-1},\ \{\mathbf{s}_t\}_{t=0}^{T} && \text{(A.2a)}\\
\text{Minimize:}\ & J\left(\{\mathbf{a}_t\}_{t=0}^{T-1},\ \{\mathbf{s}_t\}_{t=0}^{T}\right) && \text{(A.2b)}\\
\text{subject to:}\ & \mathbf{s}_0 = \mathbf{s}_{\text{start}},\ \mathbf{s}_T = \mathbf{s}_{\text{goal}} && \text{(A.2c)}\\
& \mathbf{s}_t \in S_{\text{free}},\ {}^{\forall}t \in \{0, \ldots, T\} && \text{(A.2d)}\\
& \mathbf{s}_{t+1} = F(\mathbf{s}_t, \mathbf{a}_t, \hat{\boldsymbol{\lambda}}),\ {}^{\forall}t \in \{0, \ldots, T-1\} && \text{(A.2e)}
\end{aligned}$$

J is the cost function quantifying trajectory optimality to be minimized. S_{free} denotes the set of all free states, indicating navigable space free from hazards. Solving the above problem yields the optimal action vector $\mathbf{a}_t^*$ at each t, controlling the robot to its destination. We design BenchNav to solve the given optimization problem in an iterative fashion, allowing for real-time feedback that compensates for modeling errors in F. Users can define F, J, and S_{free} to formulate the problem according to their respective objectives. BenchNav simulates navigation with deployed planning algorithms as solvers, including both global and local combinations. Here, risk inference metrics convert $\mathbb{P}(\hat{\boldsymbol{\lambda}}|\mathbf{o})$ to $\hat{\boldsymbol{\lambda}}$, incorporating uncertainty in traversability prediction. BenchNav continuously monitors the robot's state, including its experience with traversability, to observe how it safely reaches its destination within T, which is then marked as successful navigation.

A.4 Simulation Experiments

A.4.1 Implementation Details

We test BenchNav's versatility through simulations aimed at deformable off-road environments. We assume that robot motion is constrained to the terrain surface, thus defining $\mathbf{s}_{\text{start}}$ and $\mathbf{s}_{\text{goal}}$ within a 2.5D spatial space. F is then given as a unicycle model as follows:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \Delta t \cdot \begin{bmatrix} \lambda_{\text{lin}} \cdot v_t \cdot \cos(\theta_t) \\ \lambda_{\text{lin}} \cdot v_t \cdot \sin(\theta_t) \\ \lambda_{\text{ang}} \cdot \omega_t \end{bmatrix}, \tag{A.3}$$

where $\mathbf{s}_t = [x_t, y_t, \theta_t]^\top$, $\mathbf{a}_t = [v_t, \omega_t]^\top$, and $\boldsymbol{\lambda} = [\lambda_{\text{lin}}, \lambda_{\text{ang}}]^\top$, simplified as $\lambda = \lambda_{\text{lin}} = \lambda_{\text{ang}}$, link linear and angular velocities with changes in position and orientation, capturing the dynamics of robot motion. When formulating the problem, F replaces $\hat{\boldsymbol{\lambda}}$ with $\boldsymbol{\lambda}$ to form eq. (A.2e). We define J as the function that evaluates states and actions along the trajectory based on their distance to the endpoint for an efficient transition without getting stuck according to the given constraints. We here exploit $\hat{\Lambda}$ to mark S_{free} as states where $\hat{\lambda} > \lambda_{\text{stuck}}$, defining λ_{stuck} as the threshold below which a state becomes stuck due to insufficient traversability. CVaR at level $\alpha = 0.9$ quantifies the distribution to assess uncertainty incorporation in motion planning.

As problem solvers, we deploy three planning methods from distinct domains, each designed for off-road navigation.

Search-based method

Following [111], we implement a hierarchical method divided into global path planning with A* search [117] and local motion planning using the dynamic window approach (DWA) [178]. For each transition, A* finds the shortest path over long horizons in a 2.5D geometric search space, while DWA takes the resulting waypoints as subgoals to plan feasible actions over short horizons.

Sampling-based method

We use CL-RRT [164] for global motion planning that incrementally expands a tree via feedback control simulation, allowing the computation of dynamically feasible trajectories. Pure-pursuit and PID controllers manage the steering and velocities of the dynamic model, respectively. Replanning is triggered when substantial deviations occur between the actual and expected states.

Optimization-based method

We adopt MPPI [166] for navigation without the use of global path planning. While a reference path remains beneficial for long-horizon decisions, recent studies favor such waypoint-free navigation to take full advantage of planning in control space. Terminal cost is given in J to ensure long-term stability and goal alignment over an infinite horizon.

To quantify the performance of these algorithms on given problem instances, we use three key metrics.

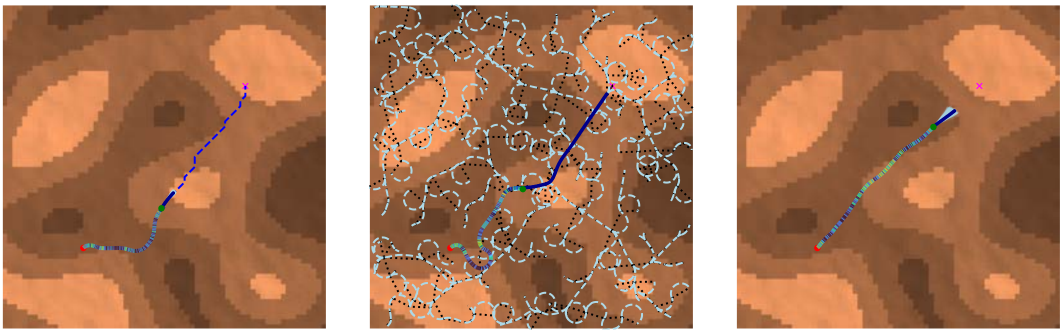
- **Success rate (Succ.)** [%] expresses the ratio of instances for which successful trajectories are found.
- **Total time ($T_{\mathbf{total}}$) [sec]** measures navigation efficiency as the total time taken to traverse a trajectory.
- **Average traversability ($\bar{\lambda}$) [%]** evaluates navigation safety by averaging all the observed traversability coefficients along a trajectory, with higher values indicating a safer traverse.

A.4.2 Experimental Setups

We prepare three environmental scenarios: standard (Std), harsh geometry (HG), and harsh shading (HS). The Std scenario adopts a simple setting mirroring the geometry and appearance trends from ML model training. The HG scenario introduces unique geometric conditions resulting in low traversability, with crater-like terrain in addition to random elevation changes. The HS scenario offers unique appearance conditions complicating traversability prediction due to ambiguous visual cues from variable shading. Each problem instance takes a $32 \times 32\,\mathrm{m}$ map with a $0.5\,\mathrm{m}$ resolution. Off-road navigation starts at $\mathbf{s}_{\mathrm{start}} = (8\,\mathrm{m}, 8\,\mathrm{m})$ and continues until $\mathbf{s}_{\mathrm{goal}} = (24\,\mathrm{m}, 24\,\mathrm{m})$.

A.4.3 Results

We highlight representative simulation examples in the Std scenario by displaying snapshots of ongoing navigation at $t = 25\,[\mathrm{sec}]$ using distinct planning methods, as illustrated in Fig. A.2. The nature of these planning algorithms is evident in their



(a) *Search-based*: A*+DWA (b) *Sampling-based*: CL-RRT (c) *Optimization-based*: MPPI

Fig. A.2: Off-road navigation snapshots with different planning algorithms at $t = 25\,[\text{sec}]$. Red circles, magenta crosses, and green circles mark the start, goal, and current robot positions, respectively. Trajectories are color-coded according to traversability, with cooler colors for safer paths. (a) A* reference paths are blue dashed lines; DWA results are shown with a dark blue solid line and light blue lines. (b) Black dashed lines show edges; light blue dashed lines are for closed-loop simulations, with solid blue lines for planned trajectories. (c) MPPI outcomes are a dark blue solid line with light blue top samples.

trajectories: Fig. A.2a shows a safe and fairly efficient trajectory under A* guidance; Fig. A.2b a suboptimal trajectory due to its sampling nature; and Fig. A.2c an efficient but slightly hazardous trajectory, a consequence of its short-horizon solutions. We emphasize BenchNav's ability to apply various planning algorithms, focusing on synthetic data generation and probabilistic traversability prediction with built-in ML models, as fundamental steps prior to motion planning. This feature allows users to easily simulate off-road navigation, facilitating problem formulation and implementation of planning algorithms while accounting for uncertain traversability.

We also present a quantitative summary of MPPI planning to see how different environmental scenarios impact off-road navigation performance. Table A.1 summarizes the results from simulation experiments across 20 problem instances for each of the Std, HG, and HS scenarios, with $T = 100\,[\text{sec}]$. MPPI performs best in the Std scenario, where ML models provide accurate traversability predictions due to training on in-domain data. The HG and HS scenarios introduce challenges resulting in lower success rates and longer traverse times as ML models struggle with out-of-domain features such as steep inclines and shaded visuals. In off-road environments, robots often face novel situations that lead to *epistemic uncertainty*, stemming from insufficient training data. Thus, incorporating uncertainty is critical in assessing the risk inherent in er-

Table A.1: MPPI planning results on different scenarios

Scenario	Succ.	T_{total}	$\bar{\lambda}$
Std	90	52.7 ± 22.5	76.3 ± 8.8
HG	80	60.2 ± 26.1	76.5 ± 4.5
HS	85	58.6 ± 25.7	74.2 ± 11.4

roneous traversability prediction. BenchNav allows us to implement various planning algorithms and explore risk inference metrics for resilient off-road navigation. It equips well-controlled environmental feature generation to replicate challenging scenarios, providing insight into the factors contributing to uncertain traversability predictions.

A.4.4 Limitations and Possible Extensions

We clarify BenchNav's limitations as 1) the lack of local observability using onboard sensors and 2) insufficient 3D representations in both perception for handling overhanging objects and motion for representing dynamic robot behaviors. We will implement these capabilities to simulate navigation algorithms designed for more dynamic, aggressive maneuvers in unknown environments [104, 168].

A.5 Conclusion

We have developed BenchNav, a simulator that tests planning algorithms alongside ML-based traversability modeling for off-road navigation. Simulation experiments verified that BenchNav consistently executes different off-road navigation algorithms and provides well-controlled datasets replicating difficulties present in traversability prediction. Future work will address the limitations described in A.4.4 to make the simulator more realistic. We also aim to extend navigation components by adopting end-to-end learning [85, 113, 114, 179] and reinforcement learning techniques [176, 180, 181].

List of Publications

Journal

[1] Masafumi Endo, Tatsunori Taniai, Genya Ishigami, "Deep Probabilistic Traversability with Test-time Adaptation for Uncertainty-aware Planetary Rover Navigation, " Under review for IEEE Access.

[2] Masafumi Endo, Genya Ishigami, "Active Traversability Learning via Risk-aware Information Gathering for Planetary Exploration Rovers," IEEE Robotics and Automation Letters, vol. 7, pp. 11855-11862, 2022.

[3] Masafumi Endo, Burak Sencer, "Accurate Prediction of Machining Cycle Times by Data-driven Modelling of NC System's Interpolation Dynamics," CIRP Annals, vol. 71, no. 1, pp. 405-408, 2022.

[4] Masafumi Endo, Shogo Endo, Kenji Nagaoka, Kazuya Yoshida, "Terrain-dependent Slip Risk Prediction for Planetary Explortion Rovers," Robotica, vol. 39, no. 10, pp. 1883-1896, 2021.

International Conference

[5] Masafumi Endo, Ashkan Jasour, Rohan Thakker, Michael Paton, Masahiro Ono, Genya Ishigami, "Real-time Integrated Task and Motion Planning under Uncertainty for Resilient Planetary Exploration: A Continuous-time Approach," In preparation for IEEE/RSJ International Conference on Intelligent Robots and Systems, Hangzhou, China, October 19-October 25, 2025.

[6] Ashkan Jasour†, Guglielmo Daddi†, Masafumi Endo†, Tiago S. Vaquero†, Michael Paton, Marlin P. Strub, Sabrina Corpino, Michel Ingham, Masahiro Ono, Rohan Thakker, "Risk-aware Integrated Task and Motion Planning for Versatile Snake

Robots under Localization Failures," Accepted for IEEE International Conference on Robotics and Automation, Atlanta, United States, May 19-May 23, 2025 († Equal contributions).

[7] Takahiro Fuke*, Masafumi Endo, Kohei Honda, Genya Ishigami, "Towards Local Minima-free Robotic Navigation: Model Predictive Path Integral Control via Repulsive Potential Augmentation," Accepted for IEEE/SICE International Symposium on System Integration, Munich, Germany, January 21-24, 2025.

[8] Masafumi Endo*, Kohei Honda, Genya Ishigami, "BenchNav: Simulation Platform for Benchmarking Off-road Navigation Algorithms with Probabilistic Traversability," IEEE International Conference on Robotics and Automation Workshop on Resilient Off-road Autonomy, 5 pages, Yokohama, Japan, May 17, 2024.

[9] Masafumi Endo*, Tatsunori Taniai, Ryo Yonetani, Genya Ishigami, "Risk-aware Path Planning via Probabilistic Fusion of Traversability Prediction for Planetary Rovers on Heterogeneous Terrains," in Proceedings of IEEE International Conference on Robotics and Automation, pp. 11852-11858, London, United Kingdom, May 29-June 2, 2023.

[10] Masafumi Endo*, Genya Ishigami, "Active Traversability Learning via Risk-aware Information Gathering for Planetary Exploration Rovers," IEEE/RSJ International Conference on Intelligent Robots and Systems, Kyoto, Japan, October 23-27, 2022.

[11] Masafumi Endo*, Burak Sencer, "Machining Cycle-time Prediction by Machine Learning of CNC Interpolator Dynamics," International Manufacturing Science and Engineering Conference, Virtual, June 21-25, 2021.

[12] Kentaro Uno*, Louis-Jerome Burtz, Masafumi Endo, Kenji Nagaoka, Kazuya Yoshida, "Quality of the 3D Point Cloud of a Time-of-flight Camera under Lunar Surface Illumination Conditions: Impact and Improvement Techniques," in Proceedings of International Symposium on Artificial Intelligence, Robotics and Automation in Space, #10a-2, Madrid, Spain, June 4-6, 2018.

[13] Toshinori Kuwahara*, Yuji Sakamoto, Shinya Fujita, Yuji Sato, Ryo Taba, Hiroto Katagiri, Masafumi Endo, Pasith Tangdhanakanond, Tomoyuki Honda, Kazuya

Yoshida, "Microsatellite Bus System Technologies of Tohoku University," IAA North East Asia Symposium on Small Satellites, Ulaanbaatar, Mongolia, August 21-23, 2017.

Domestic Conference

[14] 遠藤翔吾, 遠藤正文*, 永岡健司, 吉田和哉, "月惑星探査のためのToFカメラとトルクセンサーを用いた走行状態推定手法," ロボティクス・メカトロニクス講演会, 広島, 2019年6月5日-8日.