Uncertainty Propagation in Robot Perception

by

Parker Ewen

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Robotics)
in the University of Michigan
2025

Doctoral Committee:

Associate Professor Ramanarayan Vasudevan, Chair
Assistant Professor Maani Ghaffari
Professor Jonathan How, Massachusetts Institute of Technology
Professor Dawn Tilbury

Prior
Likelihood
Posterior
Probability Density
0.005
0.004
0.003
0.002
0.001
0.000
0.0
0.2
0.4
0.6
0.8
1.0
Coefficient of Friction
Expected Coefficient of Friction
1
0

Parker Ewen
pewen@umich.edu
ORCID iD: 0000-0002-3693-7394

© Parker Ewen 2025

DEDICATION

To my wife, family, and friends.

ACKNOWLEDGEMENTS

Writing a thesis is not a task done alone and I am thankful to everyone who contributed to making this work possible. First, I would like to thank my advisor, Ram Vasudevan, who instilled in me a strong work ethic, a love of mathematics, and a dedication to effective communication by being a role model in each of these areas. I was able to start my journey in research because you took a chance on me, and for that I am eternally grateful. You pushed me to do my best, to engage with others and share ideas, to question everything; you taught me mathematical rigor, to value myself and my work, and that research was a group effort. The lab you've put together and the team you've built are second to none. I will always remember my time in ROAHM lab, and I look forward to continuing to collaborate in the future.

I am tremendously grateful to Professors Maani Ghaffari, Dawn Tilbury, and Jonathan How for serving on my committee and providing invaluable support and feedback. To Maani Ghaffari, your help and expertise during my time at Michigan was invaluable. I am thankful we got to meet at the beginning of my PhD as our conversations together and with your students motivated much of the work presented in this dissertation. To Dawn, I appreciate that, even though you are busy running the Robotics department at Michigan, you took the time to read my thesis and provide excellent feedback. We first met when you were my CQE examiner in my first year as a PhD student, so it feels poetic that you are also on my doctoral committee as I graduate the program. Lastly, to Jonathan How, thank you for the time you took to read this dissertation and the feedback you provided, and know your past and ongoing work inspired several chapters of this thesis.

To my current labmates: Challen, Jon, Seth, Cesar, Zac, Yiting, Yizhou, Lucas, Frank, Bohao, Elena, Adam, Hao, Miracle, Sean, Austin, and Mike, and to labmates who have graduated: Patrick, Shreyas, Dan, Shannon, Matt, Sean, and JL, you are all incredible and I am thankful we got to work together. These acknowledgments would be too long if I thanked every undergraduate and graduate student that made this dissertation possible, but know that I am truly grateful for everything you all have done and I look forward to seeing where each of your careers takes you. I will always be rooting for you and hopefully one day I can help you like you helped me.

To the Robotics staff: Denise, Dan, David, Damen, Yuan, Samantha, Teresa, and others, without you we wouldn't have a department and I would have gotten lost in the bureaucracy that is a PhD years ago. Thank you for all you do and for making Robotics a supportive, welcoming, and safe place for everyone.

To the friends I made in Ann Arbor during my PhD, in Zurich during my master's, in Montreal during my undergrad, and in Ottawa before, thank you for making me laugh and smile, and keeping me sane throughout this whole process.

Thank you to my family for all your encouragement, love, and support over the years.

And finally, to my wife, who is my role model. A paragraph isn't enough to express my love and appreciation for you, so let's take this offline. You gave me the strength, love, and support to finish this dissertation and you are someone who I look to for advice on everything. I'm excited to what comes next.

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGEMENTS iii

LIST OF FIGURES viii

LIST OF TABLES xiii

ABSTRACT xiv

CHAPTER

1 Introduction 1

1.1 Motivation 1
1.1.1 Uncertainty In Geometric Models 4
1.1.2 Uncertainty In Semantic Prediction 5
1.2 Contributions 5

2 Recursive Semantic Classification and Terrain Property Estimation for Mobile Robots 8

2.1 Related Works 10
2.1.1 Geometric Mapping 10
2.1.2 Semantic Mapping 11
2.2 Preliminaries 11
2.2.1 Geometry 12
2.2.2 Sensors and Coordinate Frames 12
2.2.3 Semantic Segmentation 13
2.2.4 Probability 14
2.3 Semantic Mapping and Bayesian Inference 15
2.4 Recursive Elevation Mapping 16
2.4.1 Mesh Initialization 16
2.4.2 Point Assignment 16
2.4.3 Elevation Map Computation 17
2.5 Recursive Terrain Property Estimation 18
2.6 Implementation 19
2.7 Results 20
2.7.1 Computational Performance Evaluation 20

2.7.2 Simulation . . . 21
2.7.3 Real-World . . . 23
2.8 Extensions to Voxel Mapping . . . 25
2.8.1 Semantic Prediction as a Convolution . . . 26
2.8.2 Neural Network Comparison . . . 32
2.8.3 Simulation Results . . . 33
2.9 Extensions to Open-Vocabulary Mapping . . . 37
2.9.1 Latent Mapping Representation . . . 38
2.9.2 Latent Mapping Update . . . 40
2.9.3 Inference . . . 41
2.9.4 Uncertainty Quantification . . . 41
2.9.5 Results . . . 42
2.10 Conclusions . . . 47
3 Multi-Modal Bayesian Inference for Semantic Classification and Property Prediction . . . 49
3.1 Related Works . . . 52
3.1.1 Semantic Prediction . . . 52
3.1.2 Property Estimation . . . 52
3.1.3 Gap in the Literature . . . 53
3.2 Preliminaries . . . 54
3.2.1 Conjugate Pairs . . . 54
3.2.2 Method of Moments . . . 58
3.2.3 Revisiting the Dirichlet Normal-Inverse-Gamma Product . . . 59
3.3 Algorithm Overview . . . 61
3.4 Vision-Based Estimation . . . 62
3.5 Tactile-Based Estimation . . . 63
3.6 Implementation . . . 64
3.7 Simulation Results . . . 65
3.8 Hardware Experiments . . . 66
3.8.1 Semantic Prediction Using Property Measurements . . . 67
3.8.2 Friction Estimation Using Semantics . . . 69
3.8.3 Encoding Affordance-Based Properties . . . 71
3.9 Discussion and Conclusion . . . 73
4 Uncertainty Quantification for Radiance Field Models . . . 74
4.1 Related Works . . . 76
4.2 Methodology . . . 78
4.2.1 Radiance Fields . . . 78
4.2.2 Gaussian Splatting Models . . . 78
4.2.3 Uncertainty Quantification in Model Outputs . . . 79
4.3 Experiments . . . 81
4.3.1 Error Correlation . . . 82
4.3.2 Next-Best-View Selection . . . 83
4.4 Conclusion . . . 85

5 An Invariant-Preserving Sparse Spectral Discretization of the Continuum Equation . . . 87

5.1 Introduction . . . 87
5.1.1 Notation . . . 90
5.2 Problem Formulation . . . 90
5.3 Uncertainty Propagation for Sparse Half Densities . . . 91
5.3.1 Unitary Transformations of L^2 . . . 92
5.3.2 Hyperbolic Cross Approximation . . . 94
5.3.3 Sparse Half-Density Density Propagation . . . 95
5.4 Error Analysis on the Torus . . . 96
5.5 Implementation Details . . . 99
5.6 Numerical Experiments . . . 101
5.6.1 Benchmark $\mathbb{T}^1$ Flow . . . 101
5.6.2 Modified ABC flow over $\mathbb{T}^2$ and $\mathbb{T}^3$ Domains . . . 102
5.6.3 Harmonic Oscillator in $\mathbb{T}^6$ . . . 105
5.7 Conclusion . . . 107

6 Conclusion . . . 109

6.1 Future Directions . . . 110
6.1.1 Low-Dimensional Data Representations . . . 110
6.1.2 Active Perception Via Reinforcement Learning . . . 111
6.1.3 Fusion of Multiple Sensing Modalities . . . 112

BIBLIOGRAPHY . . . 115

LIST OF FIGURES

FIGURE

2.1 An illustration of the real-time semantic mapping method proposed in this chapter which recursively estimates terrain height and friction properties from RGB-D images. A triangular mesh represents the probabilistic estimate of the contact surface of the robot's surroundings. The proposed algorithm estimates terrain classes for each face of the triangular mesh using Bayesian inference and off-the-shelf semantic segmentation networks. Probability distributions for terrain properties are then computed (shown in the red inlay for one region). This algorithm runs on several robot systems including Boston Dynamics' Spot and Agility Robotics' Digit. 9

2.2 A flow diagram illustrating the behavior of Algorithm 1. RGB-D images are semantically segmented using an off-the-shelf semantic segmentation network. Using the camera's estimated pose and associated depth image, the pixel-wise probabilistic terrain class estimates are projected into the map. The height map is updated using a 1D Kalman filter and the terrain class estimates, alongside our novel material property dataset, are used to recursively estimate terrain properties for each region of the map. . 13

2.3 The Precision-Recall curve for terrain property estimation within the Carla simulator. We compare our method against the *Multimodal Non-Recursive* baseline for regions of high friction coefficients ($\mu > 0.5$), plotted using a solid line, and low friction coefficients ($\mu \leq 0.5$), plotted using a dashed line. Our method's performance is comparable to the baseline for regions of high friction, however, for regions of low friction our method significantly outperforms the baseline. 23

2.4 Computation times of Algorithm 1 for a 10m $\times$10m mesh with varying mesh element lengths using two off-the-shelf semantic segmentation networks, Resnet-50 [214] and Context-Encoding Resnet-50 [212], as well as the baseline algorithm assuming ground-truth semantically segmented images. The ground-truth label experiments use a pre-generated semantic segmentation image, representing the speed of Algorithm 1 without considering the time required for semantic segmentation (Line 6) and represents an lower bound on the speed of our algorithm. Error bars represent one standard deviation. 24

2.5 Each column depicts the performance of our proposed mapping algorithm (second row, the first column uses the Context-Encoding ResNet-50 trained on the Pascal dataset while the remaining columns use the RenNet-50 trained on ADE20K for terrain classification) when compared to a traversability estimation algorithm [60] (third row) applied on the scenes depicted in the top row. Traversability estimation sometimes believes that a region is traversable when it is not, such as an icy surface (Fig. 5a). In other scenarios, it believes that an area is intraversable when it is traversable such as on hills (Figs. 5b) and near low vegetation (Fig. 5c). Our method makes no claims about traversability, instead it estimates the probability distribution of terrain properties for each mesh element along with the terrain geometry. 25

2.6 Comparison of ConvBKI and Semantic MapNet on semantic mapping transferring from simulated to real data. Regions in white are unobserved by ConvBKI or in the ground truth. ConvBKI is generally successful, however suffers issues on ground segmentation due to LiDAR input. Ground segmentation is also a challenge on the simulated data, as the input semantic segmentation network is sometimes unable to distinguish sidewalk from road. 27
2.7 Comparison of ConvBKI and Semantic MapNet on BEV mapping in CarlaSC. Both networks correctly segment the driveable surface and buildings. Compared with Semantic MapNet, ConvBKI does not make predictions on unseen regions, creating some sparsity in the map. ConvBKI correctly identifies the pedestrian in the bottom right, however leaves a trace as the pedestrian walks along the crosswalk. In contrast, Semantic MapNet omits the pedestrian to achieve a higher quantitative performance. 31
2.8 Confusion matrix of ConvBKI compound on simulated BEV mapping, normalized over the predictions. ConvBKI leaves traces from dynamic objects, causing sidewalk to be incorrectly labeled as pedestrian and road as vehicle. 34
2.9 This demonstrates the overall pipeline of LatentBKI. (a) The input to LatentBKI is 3D points, which can be from LiDAR, RGB-D, or any exteroceptive sensor with 3D input. (b) Points are then processed by an off-the-shelf neural network, which encodes each point into a latent space. (c) By adopting a Gaussian likelihood over the point-wise features, we perform closed-form Bayesian inference on a voxel map where each voxel contains parameters modeling the conjugate prior of the multivariate Gaussian distribution. Additionally, instead of only considering points which fall within a voxel, we consider nearby points weighted through a kernel function. (d) The posterior predictive distribution of each voxel in latent space can then be decoded using the decoder of the neural network, enabling the computation of open-dictionary segmentation predictions with expectation and uncertainty. 37
2.10 Effect of spatial smoothing with varying levels of image sparsity. Spatial smoothing, indicated by the filter size k, is most effective for sparse images. The original image has a resolution of 720 by 1080 pixels. 45
2.11 Sparsification plot of segmentation performance compared to quantified uncertainty. As uncertain points are removed, a well calibrated uncertainty should cause the segmentation performance to increase. 46
2.12 Uncertainty maps for 5LpN3gDmAk7 MP3D sequence. (a) Covered house mesh in the sequence. (b) Categorical variance map by sampling from distribution. (c) Variance map by using E-optimality in latent space. (d) Variance map by using D-optimality in latent space. 47
2.13 Open vocabulary query task results. Query results are shown in heat map, brighter colors indicates higher values. 48

3.1 The method proposed in this chapter jointly estimates semantic classifications and physical properties by combining visual and tactile data into a single semantic mapping framework. RGB-D images are used to build a metric-semantic map that iteratively estimates semantic labels. A property measurement is taken which in turn updates both the semantic class predictions and physical property estimates. In the depicted example, the robot is unsure if the terrain in front of it is snow or ice from vision measurements alone (prior estimates) which dramatically affects the coefficient of friction and the associated gait that can be applied to safely traverse the terrain. The robot uses a tactile sensor attached to its manipulator to update its coefficient of friction estimation (posterior estimates), which then enables it to change gaits to cross the ice safely. . . . 50
3.2 A flow diagram illustrating Algorithm 5. A semantic classification algorithm predicts pixel-wise classes from RGB images that are then projected into a common mapping frame using the aligned depth image, camera intrinsics, and estimated camera pose. This semantic point cloud is used to build a metric-semantic map. When a property measurement is taken, Algorithm 4 is used to update the semantic and property estimates. 61
3.3 Results for a single simulated experiment. The image (a) and ground truth semantic labels (b) are from the Dense Material Segmentation Dataset [186]. The semantic segmentation predictions (c) do not classify parts of the desk as wood. A friction measurement is simulated using the ground-truth semantic label and Table 2.1 and is sampled from the pixel highlighted by the red cross in the RGB image. Algorithm 4 then computes the correct posterior semantic label (d). 64
3.4 A semantic segmentation task is shown and the proposed method is compared against the semantic mapping approach from [47] which is called Selmap. The expected semantic class is shown. a) The input image with measurement locations shown using an X and associated b) ground truth semantic labels are provided. Both Selmap and our approach use the same SegFormer + FastSAM pre-trained semantic segmentation network specified in Section 3.6. This network incorrectly predicts the semantic labels for several regions within the scene. c) This incorrect labelling is reflected in the Selmap implementation as the visual-based semantic mapping approach is unable to correct these erroneous predictions. By exploiting a tactile sensing modality, d) our approach is able to correct the erroneous semantic predictions and correctly predict the semantic labels of the objects within the scene. 67
3.5 The prior (dotted) and posterior (solid) friction estimates for the three ice traversal demonstration runs. When friction measurements are taken of the ice before traversal, Algorithm 4 predicts the posterior friction estimates. The friction measurements are $\psi_1 = 0.139$, $\psi_2 = 0.156$, and $\psi_3 = 0.628$ for trials 1, 2, and 3, respectively. 68
3.6 The performance of the proposed approach after a friction measurement is taken in comparison to the traversability estimate computed by [2] (bottom left) and [61] (bottom right). The proposed method is able to correctly predict that the surface in front of the robot has a low coefficient of friction which informs the locomotion planner to switch to a stable gait. In contrast, the traversability estimates predict the ice is traversable even when the robot is using a dynamic gait, which results in the robot slipping and falling. 69

3.7 The proposed method is capable of encoding affordance-based properties, in this case the force required to open a door. This door opening force is initialized as a multi-modal distribution with pushing and pulling being equally likely with a magnitude of 20N. The robot applies a pulling force of 57N on the door, resulting in it swinging open. The posterior estimate for the force required to open a door is computed and shown in the embedded plot. 71
3.8 The prior and posterior door opening force estimates running Algorithm 4 sequentially with six experiments, meaning the posterior for the previous experiment is the prior for the current experiment. The force required to open the door is indicated in the upper right-hand corner for each experiment. After measuring the force required to open 6 doors, the posterior estimate has a high variance with an expected value near $0N$, reflecting the equal likelihood that a door is either push or pull to open. 72

4.1 The ground-truth, rendered image, color variance, and color error for the Mip360 dataset. These regions of high variance correspond to regions of high rendering error, as demonstrated in Tables 4.1. Regions of high variance also correspond to regions with many translucent Gaussians and complex texture, such as grass and vegetation, indicating these regions of the image have not converged. 75
4.2 The ground-truth, rendered image, color variance, and color error for the Mip360 dataset. These regions of high variance correspond to regions of high rendering error, as demonstrated in Tables 4.1. Regions of high variance also correspond to regions with many translucent Gaussians and complex texture, such as grass and vegetation, indicating these regions of the image have not converged. 76
4.3 Pixel-wise higher-order moments and entropy are computed in the Gaussian Splatting rasterizer. The 3D Gaussian basis functions are first marginalized by projecting them onto the image plane and ordered based on distance. For each pixel, the alpha compositing process effectively computes the probability of ray termination, $p(s)$. This probability density is then used to compute the higher-order moments and entropy of each pixel. 79
4.4 The rendered variance struggles to capture un-converged regions of the splat or regions of high-frequency texture, resulting in blurry variance renders (shelf) or regions of low variance (hanging plant). Once the splat has converged, the variance likewise converges to a better approximation of the true rendering error and shows sharper, more defined regions. 84
4.5 Next-Best-View for the Lego model in the Synthetic NeRF dataset. All available training cameras are shown in gray while the training cameras chosen using our proposed variance rendering approach are shown in red. 86

5.1 This paper formulates uncertainty propagation of a density as the advection of a sparse spectral discretization of its square-root through a semi-discretized vector field, yielding a linear ordinary differential equation. The proposed approach is then able to leverage off-the-shelf ODE solvers to compute solutions for the approximate propagated uncertainty. Furthermore, this paper proves error bounds on the accuracy of these approximations as well as on the probability measure of the propagated uncertainty. . . 89
5.2 Uncertainty propagation through a 1-dimensional non-linear system with known ground-truth posterior and an initial uniform uncertainty, $\rho(x;0) \sim \mathcal{U}[0, 2\pi]$. The sparse spectral approach using our auxiliary pde maintains high accuracy while errors in the standard Galerkin approach cascade as the uncertainty is propagated. 100

5.3 The L^1-norm of the spectral discretization for the proposed sparse and standard spectral methods for the benchmark T^1 flow. The proposed approach is L^1-norm preserving whereas the standard spectral approach is not. 100
5.4 The L^p-norm errors of the standard Galerkin approach and the proposed sparse spectral method for the 1-dimensional $\mathbb{T}^1$ flow with known ground-truth. A maximum grid level of $m = 6$ is used. The proposed sparse spectral approach maintains a lower L^p-norm error compared to the standard Galerkin method for the entire experiment duration, demonstrating the utility of the auxiliary equation (5.9) in uncertainty propagation. Note, in this 1-dimensional example the hyperbolic cross is equivalent to the standard Fourier discretization. 102
5.5 The convergence of the L^p-norm errors of the propagated uncertainty at time $t = 1s$ for the 1-dimensional benchmark $\mathbb{T}^1$ flow with known ground-truth (5.31). The proposed sparse spectral approach maintains a lower L^p-norm error compared to the standard spectral method even when the same number of basis functions are used. Notably, increasing the number of Fourier basis functions does not improve the accuracy of the propagated uncertainty in the standard spectral scheme. 103
5.6 Uncertainty propagation for 1.5s through a modified ABC flow (5.32) over $\mathbb{T}^2$. a) The Monte Carlo baseline is used to compare the b) proposed sparse spectral approach and the c) standard spectral scheme. The proposed sparse spectral approach is qualitatively more accurate than the standard spectral approach even with 4.9**x** fewer basis functions. Note that the standard Galerkin approach has regions of negative probability density while the proposed sparse scheme does not. 103
5.7 Uncertainty propagation for $t = 1.5$s through the ABC flow (5.33) over $\mathbb{T}^3$ marginalized onto $\mathbb{T}^2$ for visualization. a) The baseline Monte Carlo approach is compared against the b) proposed sparse spectral method with a max level $m = 7$. Note, the sparse spectral discretization scheme uses **81x** fewer basis functions than a standard Galerkin scheme at this grid level. 105
5.8 A comparison of the average computation times and the associated three standard deviations for the a) 2D and b) 3D experiments for the proposed sparse and standard spectral methods computed using 100 trials. For the 3-dimensional system the hyperbolic approximation space results in an order-of-magnitude reduction in computation time for larger grid levels. 106
5.9 Uncertainty propagation for $t = 0.5$s for the Kuramoto model (5.34) over $\mathbb{T}^6$. A grid level of $m = 4$ is used. The marginalized, time-varying uncertainty of the first mode of the system is shown for a) the Monte Carlo approach and b) the proposed sparse grid approach. Even in 6-dimensional space the sparse spectral discretization scheme is able to run with limited computational resources with results comparable to that of Monte Carlo. 107

LIST OF TABLES

TABLE

2.1 Unimodal Gaussian parameters computed from coefficient of friction data collected across multiple terrain classes. . . . 21
2.2 The Kullback-Leibler divergence scores, average precision, and average accuracy of the two baselines and our method when applied to the Carla simulation environment. Arrows depict whether a high (↑) or low (↓) score is desired. A bolded score indicates the best performing method in each criteria. . . . 22
2.3 Results on KITTI Odometry sequence 15 [162]. As the degree of kernel expressivity increases, mIoU of ConvBKI also improves from 76.5% without optimization to 77.7% with fully optimized per-class compound kernels. . . . 30
2.4 Comparison with learning-based memory on simulated data. Metrics are reported for the isolated recurrent networks, excluding the input segmentation network. . . . 32
2.5 Per-class comparison of ConvBKI and SemanticMapNet (SMNet) on simulated data. Key results are bolded. Performance of both algorithms is similar, except on the dynamic object categories. ConvBKI leaves traces which penalizes precision of dynamic objects and recall of ground surfaces. . . . 34
2.6 Comparison against closed-dictionary BKI mapping. . . . 42
2.7 Comparison against VLMap, which performs latent mapping without spatial smoothing or uncertainty quantification. Results are compared on scenes gTV8FGcVJC9 (1) and 5LpN3gDmAk7 (2) of MP3D. . . . 44

3.1 Increase in mean pixel-wise semantic prediction accuracy using simulated property measurements and Algorithm 4. Property measurements are shown to improve semantic prediction accuracy for all material classes from Table 2.1 except snow. The standard deviation for mean pixel-wise accuracy increase is also shown in the second column. The third column shows the maximum increase in pixel-wise semantic prediction accuracy seen in the simulation experiments. . . . 66

4.1 Correlation coefficients for color variance and rendered color error. . . . 80
4.2 Correlation coefficients for depth variance and rendered depth error. . . . 80
4.3 Next best view selection on the Synthetic NeRF dataset with 10 views and the TUM dataset with 200 views. The first, second, and third best are denoted in red, orange, and yellow, respectively. . . . 82

5.1 Number of basis functions for the hyperbolic cross, $\mathbb{T}^2_m$, and the full product tensor grid, $\mathbb{T}^2_\infty$ with varying grid level m. . . . 99
5.2 Number of basis functions for the hyperbolic cross, $\mathbb{T}^3_m$, and the full product tensor grid, $\mathbb{T}^3_\infty$ with varying grid level m. . . . 99
5.3 Number of basis functions for the hyperbolic cross, $\mathbb{T}^6_m$, and the full product tensor grid, $\mathbb{T}^6_\infty$ with varying grid level m. . . . 100

ABSTRACT

Autonomous construction of shelters on Mars, robotic assistants for life-saving surgeries, and home robots integrated into our lives, once viewed as science fiction, will be attainable in the near-future thanks to advancements in robotics. For a robot to reason about its surroundings during these complex tasks, it relies on noisy sensor data. Thus, the algorithms we employ for perceptual understanding should quantify and propagate the uncertainty induced by sensor noise rather than ignore it. Additionally, robots must quantify and propagate uncertainty in real-time under computational constraints that arise due to the inherent hardware limitations of mobile robotic platforms. By quantifying and propagating perceptual uncertainty, robots can learn through their interactions with their environment and use this uncertainty for safe, robust, and reliable decision-making.

Creating these safe, robust, and reliable robotic systems necessary to accomplish these challenging tasks first necessitates the quantification of both proprioceptive (e.g. system state) and exteroceptive (e.g., metric-semantic mapping) uncertainty. Next, by propagating this uncertainty given new sensor data, a robot is capable of learning about its environment. By quantifying and propagating both proprioceptive and exteroceptive uncertainty, robots can plan more effectively and, in the case of failure or collision, can describe the factors leading to failure in a human-interpretable fashion. Quantifying and propagating uncertainty in robotics has been challenging for two reasons. First, both proprioceptive and exteroceptive models in robotics are often high-dimensional. Consequently, algorithms designed to handle such high-dimensional models must be efficient while managing finite computational memory to operate in real-time. Second, real-world systems are nonlinear and their associated models of uncertainty are non-Gaussian, making tractable closed-form propagation challenging.

This dissertation tackles the challenges associated with uncertainty in robotic perception by proposing a framework for uncertainty propagation for both proprioceptive and exteroceptive models. In particular this dissertation focuses on (1) uncertainty quantification and propagation for robotic metric-semantic maps representing exteroceptive sensing and (2) uncertainty propagation over Riemannian manifolds, which play a key role in robotic proprioception. These contributions allow roboticists to take an important step towards not only quantifying but also propagating uncertainty in perceptual data for robotic systems in real-time while operating with limited computational resources.

CHAPTER 1

Introduction

1.1 Motivation

Perception is one of three central pillars of robotics, alongside planning and control, and encompasses both intrinsic and extrinsic perception. Intrinsic perception is responsible for sensing internal quantities of the robot itself, such as joint angles, center-of-mass acceleration, or temperature. In contrast, extrinsic perception is responsible for sensing quantities of a robot's environment, such as obstacle locations, terrain properties, or the location of the robot in the world. Both intrinsic and extrinsic perception are necessary for downstream tasks in robotic planning and control [211], and the robustness and reliability of perceptual algorithms thus directly correlates to the robustness and reliability of these tasks.

Perception algorithms require sensors to capture and record information about a robot and its surroundings. Joint encoders, inertial measurement units, and thermistors, among others, capture intrinsic quantities [210] while cameras, lidar, sonar, and other such devices capture information about a robot's surroundings. Unfortunately, these sensors are often noisy, meaning the true measured quantity is often corrupted by some form of noise such as sensor drift, thermal noise, or electrical noise [141]. To effectively process the stream of data coming from these sensors, perceptual algorithms must confront a significant challenge: the inherent unreliability of these sensors due to noise. This noise can distort the data and complicate interpretations. This unreliability cascades to downstream planning and control tasks, and can cause robotic systems to fail [155].

Robotic perception has seen a surge in capabilities in the past decade thanks in part to the advent of modern machine learning algorithms [63, 138]. The broad availability of data and introduction of foundational models has enabled advancements in scene reconstruction, object identification, and mapping [103, 204]. The success of these methods lies in the ability of neural networks to decode complex patterns in perceptual data and adapt to various tasks which has propelled the development and capabilities of robotic perceptual systems [77].

Machine learning methods used in robot perception often fall into the category of *discriminative* models, a superset of regression and classification models [136]. Discriminative models are generally considered deterministic where, given the same inputs, they will return the same outputs; this feature is highly valued as it is what enables these networks to be trained in the first place. When deployed in the real-world, learning-based discriminative methods must contend with the aforementioned noisy sensor measurements and, even more challenging, unpredictable agents such as pedestrians or other motor vehicles.

Coupled with the fact that machine learning methods are heavily reliant on the data used for training, these challenging and noisy real-world environments often degrade the performance of machine learning methods, sometimes with catastrophic results. Many such learning-based approaches disregard sensor noise to simplify computation. By disregarding the presence of noise, however, many perception algorithms may make incorrect predictions [47]. This can result in autonomous vehicles colliding with cars or pedestrians, walking robots falling on slippery ground, or drones crashing into the environment. In situations like these, where precision is essential yet difficult to guarantee due to the perpetual uncertainty resulting from sensor noise, deterministic machine learning models fall short; they fail to fully capture the uncertainty inherent to perception.

Another challenge in addressing perceptual uncertainty in robotic systems is dealing with how this uncertainty propagates or changes over time. For instance, an autonomous vehicle that is uncertainty about the location of a pedestrian should track how that uncertainty evolves if the pedestrian were to walk through a crosswalk [208]. Uncertainty propagation is vital in robotics because it enables the design of algorithms and systems that can operate reliably despite the inherent perceptual uncertainty of real-world scenarios [111].

Uncertainty and uncertainty propagation are also key considerations for robots operating in dynamic and unstructured environments. A robot performing a task such as search and rescue, agricultural monitoring, or planetary exploration must adapt to changes in its environment and unexpected events. Uncertainty propagation enables these robots to assess the likelihood of their perceptual data and consider this likelihood in their decision-making processes and update their plans accordingly [23]. This adaptability is necessary in ensuring task completion, preventing failures, and avoiding potentially dangerous situations [46]. Uncertainty propagation also plays an essential role in the domain of human-robot interaction. To interact safely with humans, evaluating uncertainty in human pose estimation, human motion, and human intention while also propagating this uncertainty through a series of actions or movements is necessary to prevent human-robot collisions or other potential harm towards a human [75].

Moreover, the ability to handle and propagate uncertainty has implications on a robot's

autonomy and its need for human intervention. A robot that can accurately model uncertainty of its environment can make better decisions about when to confidently act autonomously and when to seek human assistance [46]. This is especially important in scenarios where human intervention may be difficult or slow to obtain, such as in space exploration or deep-sea missions. In such cases, accurate uncertainty propagation can mean the difference between mission success and failure. Furthermore, by considering uncertainty in sensor data and propagating this uncertainty through time, robots may also autonomously take actions which reduce uncertainty [91]. For example, a robot which is uncertain about the location of a particular far away object may decide to approach the object to estimate its location more precisely.

To address the challenge of sensor noise and uncertainty propagation in perception, early robotics research took a probabilistic approach and enabled the estimation of perceptual uncertainty [182]. These early probabilistic approaches enabled robot perception algorithms to incorporate and propagate simplified sensor noise and uncertainty models. These simplifications often took the form of linear and Gaussian assumptions [192]. Accounting for uncertainty in downstream tasks was then made possible given its tractable modelling in the perceptual pipeline.

These linear and Gaussian approximations enable exact uncertainty propagation, but the increasing complexity of robotic tasks and the general nonlinearity of robot systems result in the degrading performance of these approximations [96]. This poses a further challenge; not only do robots need to contend with perceptual uncertainty, this uncertainty must be propagated through potentially nonlinear systems.

The advantages of probabilistic modeling is that it offers a perspective that accommodates the noisy and uncertain nature of the real world. This, however, introduces the challenge of propagating uncertainty through robot models, a particularly challenging task for nonlinear systems. As a robot navigates and perceives its environment, nonlinearity and non-Gaussian uncertainty resulting from noisy sensor data, environmental dynamics, and action models cause conventional uncertainty propagation techniques to struggle or be intractable.

The complexity of propagating uncertainty may lead some to question its necessity. While challenging, incorporating uncertainty into the downstream pipeline of planning and control enables safety, reliability, and robustness in these downstream tasks [120]. It is therefore desirable to propagate uncertainty in perceptual algorithms.

The central aim of this dissertation is to address uncertainty propagation for robotic perception. Of primary concern is the propagation of semantic classification uncertainty in robot mapping and the subsequent uncertainty in an object's physical properties. This dissertation presents methods for single and multiple sensing modalities which are capable

of generating accurate semantic and property predictions while maintaining uncertainty in these predictions. To complement semantic and property prediction uncertainty, this dissertation also presents a means of quantifying geometric uncertainty for radiance field methods. These methods correspond to uncertainty propagation in extrinsic perception. Finally, this dissertation also presents a novel sparse spectral discretization scheme for the propagation of non-Gaussian uncertainty through nonlinear systems, which is aimed at addressing uncertainty propagation in intrinsic perception for robotic systems.

1.1.1 Uncertainty In Geometric Models

Geometric models are foundational to many robotic tasks, serving as the basis for perception, navigation, and manipulation. These models enable robots to interpret and interact with their surroundings, representing objects and surfaces through simplified structures such as point clouds, meshes, voxels, or radiance field models [106]. By leveraging geometric representations, robots can generate path plans, perform object manipulation [20], maintain situational awareness in dynamic environments, and avoid collisions [122]. Through sensor input, these models can also be updated in real-time, allowing robots to adapt to changing conditions and perform tasks autonomously or in collaboration with humans [30].

In robotics, incorporating uncertainty into geometric models is crucial because the data from sensors, like LiDAR, cameras, or inertial measurement units (IMUs), are often noisy or incomplete. Environmental factors such as lighting changes, occlusions, and sensor limitations can introduce further inaccuracies into geometric representation. These uncertainties can propagate to downstream tasks, affecting the robot's ability to perform reliable state estimation, control, and decision-making. By accounting for uncertainty in geometric models, robots can assess the reliability of their perceptions and actions, or decide that more data is needed in order to accomplish a task.

In Chapters 2 and 4 I describe an approach for modelling geometric uncertainty in a contact surface mesh and a radiance field model, respectively. For the contact surface mesh, a 2.5D surface composed of connecting triangles, the height of each triangle vertex is treated as a Gaussian random variable. Bayesian inference is then used to update the likelihood of these vertex heights, where geometric observations in the form of depth camera point clouds are used. For radiance field models, we approximate the rendering function as a probabilistic process, treating each pixel as an independent random variable. When rendering depth, the pixel-wise variance of the rendering is computed in a straightforward manner and provides a means of quantifying the uncertainty of the radiance field model output. The quality of these geometric uncertainty quantification approaches is demonstrated via their correlation

with model error and we demonstrate improvements over state-of-the-art methods.

1.1.2 Uncertainty In Semantic Prediction

Semantic predictions enable robots to understand not just the geometric structure of their environment but also the function of objects within it. Through techniques like semantic segmentation, object detection, and classification, robots can assign labels to different elements in their surroundings, distinguishing between obstacles, landmarks, and interactable objects like doors or tools. This semantic understanding is crucial for many robotics applications, such as autonomous navigation, object manipulation, and human-robot interaction, where the robot must interpret its environment in a meaningful way to accomplish complex tasks.

However, akin to geometric models, semantic predictions are inherently uncertain due to limitations in the robot's perception system. Sensor noise, incomplete data, varying environmental conditions, and stochasticity in machine learning models used for semantic interpretation can all introduce uncertainty. For instance, changing lighting conditions, overlapping objects, or occlusions can cause a robot to misclassify objects, which may lead to incorrect actions, unsafe behavior, or task failure. Managing uncertainty in semantic predictions is therefore vital for ensuring the robustness and reliability of robotic systems.

Incorporating uncertainty into semantic predictions enables a robot to assess its confidence in its understanding of the environment, leading to more informed decision-making. Instead of relying on a single, potentially flawed prediction, a robot can weigh multiple possibilities based on their likelihood. This is particularly useful in dynamic or cluttered environments where ambiguity is common.

In Chapters 2 and 3 I describe an approach for modeling semantic uncertainty by projecting semantic classifications onto a geometric surface and performing Bayesian inference over the geometric surface elements. We assume that pixel-wise semantic observations are in the form of a Categorical distribution and exploit the Dirichlet conjugate prior to efficiently compute the posterior belief for the semantics of each geometric surface element. We demonstrate that this approach results in improved semantic prediction performance compared to state-of-the-art approaches and enables multiple sensing modalities, such as vision and touch, to be used in tandem to better predict semantic classes.

1.2 Contributions

The focus of this dissertation is on the computationally tractable propagation of uncertainty for robotic perception. Each of the presented contributions is validated either in simulation,

on hardware, or both. This dissertation consists of four contributions; the first three correspond to modeling uncertainty propagation for semantic classifications and physical property estimates in extrinsic perception, while the final contribution corresponds to propagating uncertainty in compact domains and is primarily motivated by intrinsic perception.

The first contribution in Chapter 2 is a computationally tractable means of propagating semantic classifications and conditioned property estimates using captured image data. Semantic classification via machine learning methods is widely used in robotics, so the proposed approach is widely applicable. Physical properties are then conditioned on these recursively estimated semantic classifications and the efficacy of the combined semantic and property prediction algorithm is validated against state-of-the-art algorithms from the mapping literature. The contribution is applied in the context of elevation mapping for legged robotics to demonstrate its utility and originated in [47].

The second contribution in Chapter 3 builds off the first and introduces a means of exploiting multiple sensing modalities to increase the accuracy of the semantic classifications and property predictions. Whereas the previous contribution conditioned physical properties on semantic classifications, this algorithm introduces the opposing relationship, where semantic classifications are conditioned on physical properties. This is motivated by the fact that material properties such as the coefficient of friction can inform material classifications; ice is slippery while concrete is rough. A tactile sensor is used to measure physical properties and the proposed algorithm uses these measurements to update the semantic classification likelihood and property estimates. This work originally appeared in [44].

The third contribution in Chapter 4 demonstrates how to estimate radiance field model uncertainty during rendering. Rather than estimating model parameter uncertainty and translating this into model output uncertainty, we directly quantify uncertainty of the model by treating the rendering function as a probabilistic process for each pixel. We show that the pixel-wise variance and entropy computed in this manner have a stronger correlation with the rendering error than existing state-of-the-art radiance field uncertainty quantification approaches. Furthermore, we also demonstrate that the rendered uncertainty may be used as a training signal, helping to improve the accuracy of the radiance field model by tailoring the training process to focus on regions of high uncertainty.

Finally, *the fourth contribution* in Chapter 5 introduces uncertainty propagation through the lens of solving linear partial differential equation. This linear partial differential advection equation is presented and a sparse spectral discretization scheme is derived for the propagation of probability measure through arbitrary smooth system dynamics. Bounds on the approximation error for the propagated probability measure are derived and the proposed sparse spectral framework is validated in simulation.

Finally, concluding remarks are provided in Chapter 6 alongside a discussion of future work.

CHAPTER 2

Recursive Semantic Classification and Terrain Property Estimation for Mobile Robots

Mapping from images or point clouds enables mobile robots to perform object avoidance and terrain traversal [114]. Prior work in mapping for mobile robots has focused on generating maps by reconstructing the surface geometry in the vicinity of the robot as a 2.5-D polygonal mesh [49], often referred to as an elevation map or contact surface. These representations describe the geometry of the robot's surroundings, but have no information regarding the properties of the underlying terrain such as the friction coefficient. The equations of motion governing the behavior of mobile robots are a function of both the internal state of the robot and the properties of the terrain over which the robot is traversing [128, 48]. As a result, maps used for robot navigation should include information about these properties. Work has been done in estimating terrain properties such as friction or the internal shear coefficients of granular surfaces from single RGB images [129, 132, 19]. Building semantic maps with terrain property estimates using these methods has remained a challenge due to the difficulty of incorporating prior information into these maps.

As illustrated in Fig. 2.1, the contributions of this chapter are two-fold. First, we develop a novel dataset that is used to model the relationship between the coefficient of friction and a variety of terrain classes. Second, we propose a robot-centric semantic mapping framework by which geometric and terrain properties are estimated using a closed-form Bayesian inference algorithm.

The maps generated by this algorithm map 5m around the robot with discretizations of 2cm and are computed at speeds between 9Hz $\pm$5Hz. The proposed mapping algorithm is accessible via an open-source ROS package which takes RGB-D images and camera pose mean and covariance estimates as inputs and outputs the semantic map. We use our semantic mapping framework to estimate the coefficient of friction both in simulation and in the real-world and show that it outperforms existing methods from the literature in terms of accuracy in property estimation.

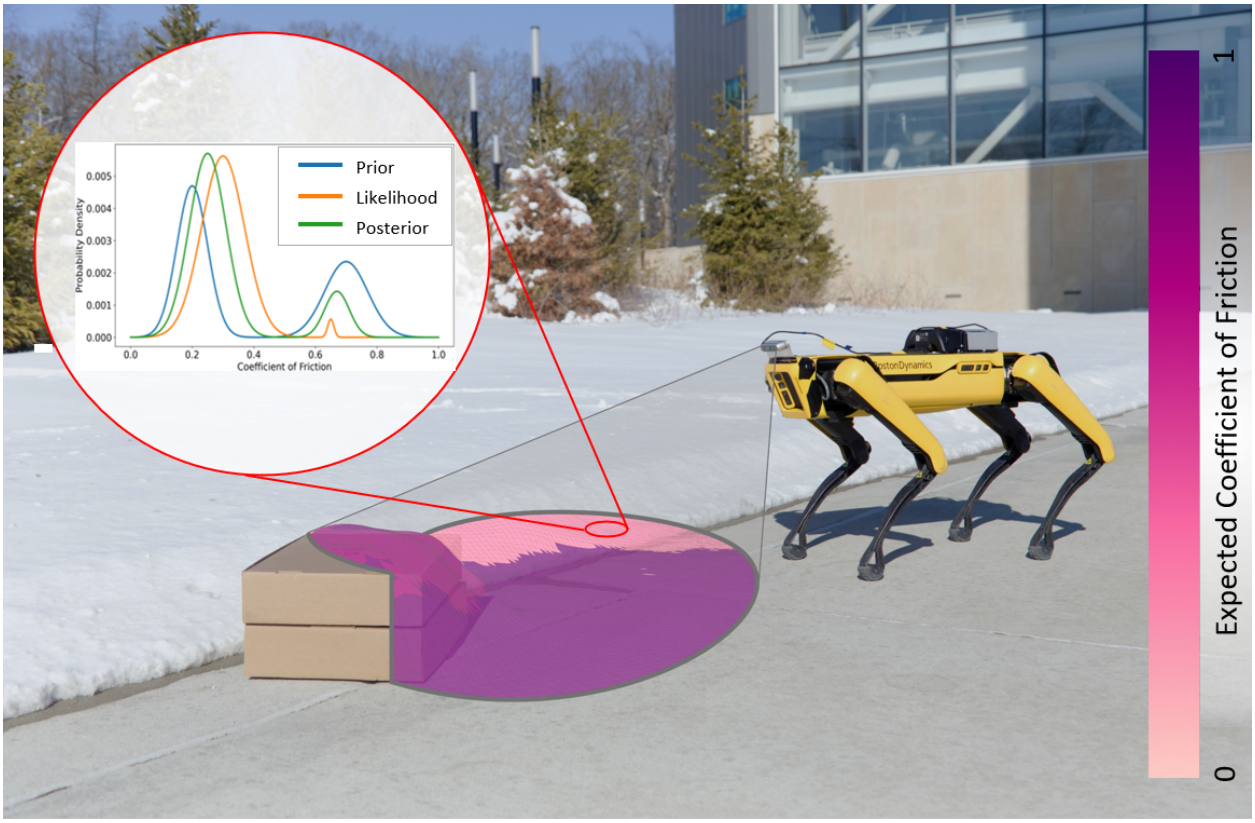


Figure 2.1: An illustration of the real-time semantic mapping method proposed in this chapter which recursively estimates terrain height and friction properties from RGB-D images. A triangular mesh represents the probabilistic estimate of the contact surface of the robot's surroundings. The proposed algorithm estimates terrain classes for each face of the triangular mesh using Bayesian inference and off-the-shelf semantic segmentation networks. Probability distributions for terrain properties are then computed (shown in the red inlay for one region). This algorithm runs on several robot systems including Boston Dynamics' Spot and Agility Robotics' Digit.

The remainder of this chapter is organized as follows: Section 2.1 summarizes the mapping literature and Section 2.2 introduces preliminary concepts used throughout the chapter. Section 2.3 provides an overview of our method and Sections 2.4 and 2.5 pertain to the recursive elevation mapping and recursive terrain property estimation portions of our algorithm, respectively. Section 2.6 discusses implementation details and describes our dataset used to model the coefficient of friction for ten terrain classes. Section 2.7 describes the evaluation of our algorithm both in simulation and in real-world scenarios. Section 2.10 follows with concluding remarks and a discussion on future work.

2.1 Related Works

This section describes the existing literature on geometric and semantic mapping methods with an emphasis on algorithms pertaining to legged locomotion. Geometric maps represent the interface between free-space and occupied space, and are used to model the environment around the robot. Semantic maps include information such as terrain class, terrain properties, or additional high-level labels within the representation. Often semantic maps include a geometric representation onto which the semantic information is projected, and thus in these cases, they may contain more information than geometric maps.

2.1.1 Geometric Mapping

A robot's surroundings can be geometrically modelled by constructing representations of the underlying terrain surface using range sensor data. These range sensors produce sparse point cloud representations, which must then be converted into continuous or piecewise structures to be of use for planning. We organize prior work in geometric mapping into either volumetric or 2.5-D piecewise-planar categories.

Volumetric representations map the 3D geometry of a scene using discretized volumes, such as voxelized occupancy grids, or via Truncated Signed Distance Fields (TSDF) or Euclidean Signed Distance Fields (ESDF). Voxelized occupancy maps, for instance, are the three dimensional equivalents to occupancy grid maps. Voxels are given a binary label representing either free-space or occupied space, and the probability of each label given new data is updated by applying a Bayes filter. Such volumetric representations are often memory and computational resource intensive, but one may address some of these limitations by applying memory-efficient representations such as octrees [83]. Other representations, like TSDFs and ESDFs, map the distance to the nearest occupied cell rather than storing the entire volumetric representation of the environment [135].

Often for legged robotic locomotion and footstep planning, only a lower-dimensional rather than volumetric representation is required. Point-foot robots, such as Boston Dynamics's Spot, require maps with discretizations on the order of 1cm to perform footstep planning due to their footprint size [49]. State-of-the-art volumetric representations with 1cm resolution are only able to operate at around 1Hz [83].

The most common geometric mapping paradigm in legged robotics is the elevation map, where a piecewise-planar mesh is used to represent the terrain [114, 49]. Such maps also have parallels in the surface reconstruction community [215]. Recently, Bayesian inference has been applied to recursively update an elevation map to minimize the impact of noise from the range sensor measurements and robot pose estimation [49]. Such methods run at 20-100Hz,

and have been used for legged locomotion and footstep planning [114]. One shortcoming of geometric mapping techniques is that they fail to account for terrain properties, which are an essential component in the stability of a legged robotic platform [128]. To compliment geometric mapping techniques, it is important to include information on terrain properties via semantic mapping.

2.1.2 Semantic Mapping

Semantic mapping is a broad field since high-level labels are task dependent. Methods predicting terrain properties are of interest for robotic navigation, and we limit our focus within semantic mapping to such methods. Semantic labels or attributes can be represented in abstract topological layers [104], but it has become commonplace for semantic information to be estimated from images or point clouds using neural networks and then projected onto geometric representations [60]. Visual information has been previously applied to predict the expected value for the coefficient of friction on roads [194] and other common terrain types [19, 132], as well as slip predictions for wheeled robots [5]. Current semantic mapping methods are non-recursive, meaning they cannot use priors to refine estimates, and as such, these methods are not robust while dealing with noisy sensor data.

One common semantic mapping paradigm is traversability estimation. Traversability mapping has been used to bypass the need to estimate terrain properties by instead estimating which regions in the environment a robot can traverse [60, 138]. Such mapping strategies often fail to consider that traversability is a function of the robot's internal state, such as the acceleration of the robot's center-of-mass or wheel velocity, and as such, traversability estimation methods often over- or under-approximate traversable regions [99]. Recent work has shown how to incorporate traversibility classifications into a voxel occupancy grid map using a Bayesian inference framework to update traversibility estimates based on new observations [60]. While this represents a large step towards a recursive framework for semantic mapping, it fails to address the short-comings associated with the robot-dependent nature of traversability.

2.2 Preliminaries

This section introduces notation, geometric concepts, coordinate frames, semantic segmentation, and probability theory used within this chapter.

Vectors, written as columns, are typeset in bold and lowercase, while sets and matrices are typeset in uppercase. The element i of a vector $\boldsymbol{x}$ is denoted as x_i. An n-dimensional

open (resp. closed) interval is denoted by $(a, b)^n$ (resp. $[a, b]^n$). The power set of a set $\mathcal{A}$ is denoted as $2^{\mathcal{A}}$. Throughout the chapter we let f refer to a probability mass or density function.

2.2.1 Geometry

Barycentric coordinate systems specify the location of a point with respect to the vertices of a b-dimensional simplex. Given a point $\boldsymbol{p} \in \mathbb{R}^a$ in Euclidean space, we compute the corresponding b-dimensional Barycentric coordinate representation $\boldsymbol{\lambda} \in \mathbb{R}^b$ using a change of basis function $g : \mathbb{R}^a \to \mathbb{R}^b$ [213], with $a \geq b$. We test whether a point $\boldsymbol{p}$ lies within a b-dimensional simplex using the Barycentric coordinates by the following theorem:

Theorem 1. *Let $\boldsymbol{p} \in \mathbb{R}^a$ be a point in Euclidean space and let $\boldsymbol{\lambda} \in \mathbb{R}^b$ be its corresponding Barycentric coordinates. Point $\boldsymbol{p}$ lies within the simplex if and only if $\lambda_i \in (0, 1)$ for all λ_i in $\boldsymbol{\lambda}$.*

2.2.2 Sensors and Coordinate Frames

To simplify exposition, we assume the following:

Assumption 2. *An RGB-D camera that is attached to a robot with known camera intrinsics is used to collect data.*

These RGB-D images, camera intrinsics, and an estimated camera pose are used to compute projected point clouds within the camera frame. This process is discussed here as well as in Section 2.2.3. This subsection introduces the notation used to describe coordinate frames and transformations between coordinate frames. Note that one could extend the algorithms presented in this chapter to multiple cameras or even LiDAR; we focus on a single RGB-D camera for simplicity.

We denote a vector from the origin to point $\boldsymbol{p}$ in coordinate frame A as $\boldsymbol{p}_A$. The rotation matrix from coordinate frame A to B parameterized by quaternions $\boldsymbol{q}$ between the two frames is denoted as $R_A^B(\boldsymbol{q})$. Three coordinate frames are used within the chapter: the world frame W, the sensor frame S, and the mapping frame M. The inertial frame is space-fixed and the environment is assumed to be static with respect to this frame. The sensor frame S is located at the center of the camera with the z-axis pointed out of the camera into the scene. The transformation from the robot's center-of-mass to the sensor frame is static, and we assume that this transformation is known. Lastly, the mapping frame M is defined in relation to the location of the robot. Its origin corresponds to the robot's center-of-mass

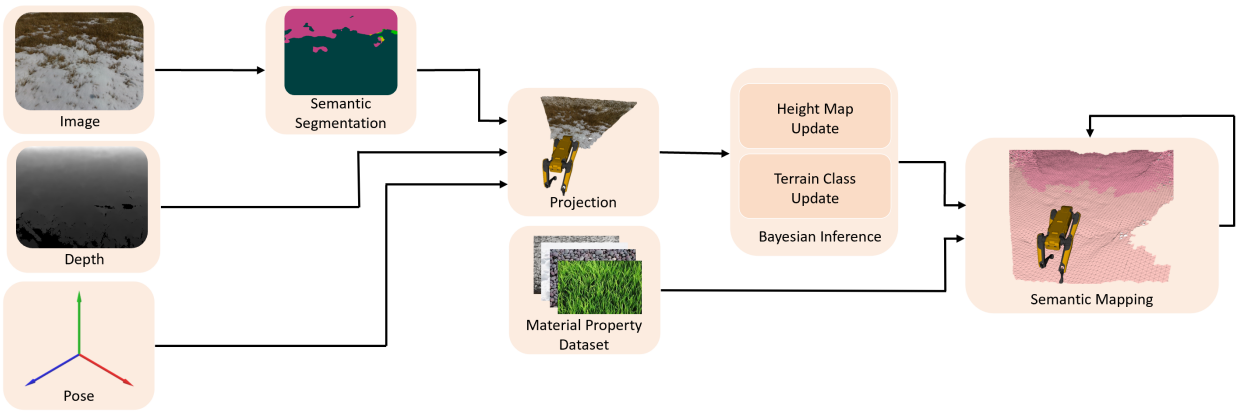


Figure 2.2: A flow diagram illustrating the behavior of Algorithm 1. RGB-D images are semantically segmented using an off-the-shelf semantic segmentation network. Using the camera's estimated pose and associated depth image, the pixel-wise probabilistic terrain class estimates are projected into the map. The height map is updated using a 1D Kalman filter and the terrain class estimates, alongside our novel material property dataset, are used to recursively estimate terrain properties for each region of the map.

projected onto the terrain, the x-axis (resp. y-axis) is oriented towards the front (resp. left-side) of the robot, and the z-axis is aligned with the z-axis of the inertial frame. Given the vector to any range sensor measurement point in the sensor frame, $\boldsymbol{p}_S$, we transform it into the mapping frame via an affine transform:

$$\boldsymbol{p}_M = R_S^{M\intercal}(\boldsymbol{q}) \cdot \boldsymbol{p}_S - \boldsymbol{t}_S^M \tag{2.1}$$

where $\boldsymbol{t}_S^M$ represents the translation from the sensor frame to the map frame.

2.2.3 Semantic Segmentation

Semantic segmentation assigns class probability scores to each pixel in an image. The classes in semantic segmentation are task-dependent. This chapter focuses on the list of terrain classes described in Table 2.1.

It is common to use convolutional neural networks to estimate pixel-wise class probability scores [63]. Let $I \in \mathbb{R}^{w \times h \times 3}$ denote an RGB image, where h, w are the height and width of the image in pixels. A trained semantic segmentation network takes an input image and outputs the pixel-wise terrain class probability scores $T \in \mathbb{R}^{w \times h \times k}$ for k terrain classes in the form of a k-dimensional Categorical Distribution. The accuracy of the semantic segmentation depends on the network used. Note, this is not the emphasis of this chapter. We use the aligned depth image $D \in \mathbb{R}^{w \times h}$ to project the pixel-wise terrain class probability scores T

into the sensor frame using the camera intrinsics and the camera projection equation [131, (10.38)] to obtain a point cloud representation of the semantically segmented image.

2.2.4 Probability

This subsection reviews the Categorical Distribution, the Dirichlet Distribution, and their relation to one another. We denote a random variable as z or the vector of random variables as $\boldsymbol{z}$. The Categorical Distribution is a discrete k-dimensional distribution parameterized by a vector $\boldsymbol{\theta} \in [0,1]^k$. The probability mass function of the Categorical Distribution represents the probability that sample z belongs to class i, where $i \in \{1, 2, \ldots, k\}$:

$$f(z = i|\boldsymbol{\theta}) = \theta_i \tag{2.2}$$

The Dirichlet Distribution is a continuous k-variate probability distribution which is parameterized by a vector $\boldsymbol{\alpha} \in \mathbb{R}^k_{\geq 0}$ of positive reals. The probability density function of the Dirichlet Distribution is defined below:

$$f(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{j=1}^k \alpha_j)}{\sum_{j=1}^k \Gamma(\alpha_j)} \prod_{j=1}^k \theta_j^{\alpha_j - 1} \tag{2.3}$$

where

$$\Gamma(\alpha_j) = \int_0^\infty x^{\alpha_j - 1} \exp(-x) dx. \tag{2.4}$$

Suppose we obtain n measurements $\mathcal{Z} = \{z_1, \ldots, z_n\}$ of a given region, represented as random variables drawn from a Categorical Distribution. Our goal is to apply Bayesian inference to predict the probability that a new measurement of the same region belongs to terrain class i given prior measurements $\mathcal{Z}$. That is we want to compute $f(z = i|\mathcal{Z}, \boldsymbol{\alpha})$. Note, that we have assumed for full generality that $f(z|\mathcal{Z}, \boldsymbol{\alpha})$ is a function of some hyperparameters $\boldsymbol{\alpha}$. To do this, one could compute $f(z|\mathcal{Z}, \boldsymbol{\alpha}) = \int_{\boldsymbol{\theta}} f(z|\boldsymbol{\theta}) f(\boldsymbol{\theta}|\mathcal{Z}, \boldsymbol{\alpha}) \mathrm{d}\theta$, but this would require constructing $f(\boldsymbol{\theta}|\mathcal{Z}, \boldsymbol{\alpha})$. By applying Bayes Theorem, one can write

$$f(z|\mathcal{Z}, \boldsymbol{\alpha}) = \int_{\boldsymbol{\theta}} f(z|\boldsymbol{\theta}) \frac{f(\mathcal{Z}|\boldsymbol{\theta}, \boldsymbol{\alpha}) f(\boldsymbol{\theta}|\boldsymbol{\alpha})}{f(\mathcal{Z}|\boldsymbol{\alpha})} \mathrm{d}\theta. \tag{2.5}$$

Generally, this integral is hard to compute exactly.

To compute a closed form expression for $f(z|\mathcal{Z}, \boldsymbol{\alpha})$, we use the notion of conjugate prior [184]. In particular, we choose to represent $f(\boldsymbol{\theta}|\boldsymbol{\alpha})$ as a Dirichlet Distribution, which is the conjugate prior to the Categorical Distribution $f(z|\boldsymbol{\theta})$. With this choice, one can prove that

Algorithm 1: Recursive Semantic Mapping

$\mathcal{G} \leftarrow$ groundPlane() // Sec. 2.4.1
$\mathcal{V}$ collection of vertices // Sec. 2.4.1
$\Xi \leftarrow$ triangulation($\mathcal{V}$) // Sec. 2.4.1
while robot is running **do**
  $I, D, \boldsymbol{q} \leftarrow$ getImageAndSensorPose()
  $T \leftarrow$ semanticallySegmentImage(I)
  $P_M \leftarrow$ projectImage($T, D, \boldsymbol{q}$) // Sec. 2.2.3
  assign points $\boldsymbol{p}_M \in P_M$ to $\xi \in \Xi$ // Alg. 2
  $\mathcal{V} \leftarrow$ updateElevationMap($\mathcal{V}, \Xi$) // Alg. 3
  $\Xi \leftarrow$ updateTerrainPrediction(P, Ξ) // Sec. 2.5

$f(\boldsymbol{\theta}|\mathcal{Z}, \boldsymbol{\alpha})$ is also a Dirichlet Distribution parameterized by a vector $\tilde{\boldsymbol{\alpha}}$:

$$\tilde{\alpha}_j = \alpha_j + \sum_{z_i \in \mathcal{Z}} 1\{z_i = j\}, \tag{2.6}$$

where $1\{z_i = j\}$ is equal to 1 when the expected terrain class of measurement z_i is class j and is zero otherwise [184]. By using this property in (2.5), one can prove [184, (3)] the probability that a new measurement of the same region belongs to terrain class i given prior measurements $\mathcal{Z}$ is:

$$f(z = i|\mathcal{Z}, \boldsymbol{\alpha}) = \frac{\tilde{\alpha}_i}{\sum_{j=1}^{k} \tilde{\alpha}_j}. \tag{2.7}$$

2.3 Semantic Mapping and Bayesian Inference

As illustrated in Fig. 2.2, this section summarizes our robot-centric semantic mapping algorithm used to estimate the terrain surface profile and properties using a triangular mesh representation given an RGB-D camera with known pose (Algorithm 1). Subsequent sections describe each step of Algorithm 1 in detail. The mesh is described using two collections. The first is the collection $\mathcal{V} \subset (\mathbb{R}^4)^m$ of vertices $\boldsymbol{v} = [v_x, v_y, v_z, v_{\sigma^2}]$, where m is the number of vertices within the mesh. The first three components of a vertex, v_x, v_y, and v_z, correspond to the Euclidean position of the vertex with respect to the mapping frame M, and the last component v_{σ^2} corresponds to the variance of v_z. The second is the collection $\Xi \subset \mathcal{V}^3 \times (\mathbb{R}^{3+k})^l \times \mathbb{R}^k_{\geq 0}$ of mesh elements, or faces, ξ. An element ξ is a collection of three components: the three vertices whose connecting line segments define the perimeter of the face, interior points, and a vector of Dirichlet parameters. The interior points are discussed in Section 2.4.2.

We start by defining a flat ground plane $\mathcal{G}$ with zero height (Line 1). Next, vertices $\boldsymbol{v} \in \mathcal{V}$ and mesh elements $\xi \in \Xi$ are initialized (Lines 2-3, Section 2.4.1). We retrieve the RGB-D image, I and D, and camera pose estimate in the world frame, $\boldsymbol{q}$, from the robot (Line 5). A semantic segmentation network takes the RGB image I and outputs pixel-wise terrain class probability scores, T (Line 6). These pixel-wise scores are projected into the mapping frame (Line 7, Section 2.2.2) and assigned as interior points to a mesh element ξ (Line 8, Section 2.4.2). Interior points are used to compute vertex heights, vertex height covariance, and terrain labels of the corresponding mesh element ξ. The height map is updated using the projected points (Line 9, Section 2.4.3). and terrain properties are recursively updated via the Dirichlet-Categorical conjugacy relationship (Line 10, Section 2.5).

2.4 Recursive Elevation Mapping

2.4.1 Mesh Initialization

At startup, we define a flat ground plane $\mathcal{G}$ with zero height (Line 1) and initialize a grid pattern of evenly-spaced vertices $\boldsymbol{v} \in \mathcal{V}$ with zero height and zero variance (Line 2). The set of faces $\xi \in \Xi$ are initialized (Line 3) by triangulating these vertices into a set of equal-sized, isosceles, right-angled triangles. The set of interior points of each face is initialized as an empty set and the Dirichlet parameters are initialized as a vector of zeros.

2.4.2 Point Assignment

Interior points represent the set of points $\boldsymbol{p}_M$ whose projection lie within the 2-dimensional simplex defined by the perimeter of ξ. The process by which points are projected and assigned as interior points is described in Alg. 2. The camera on the robot collects RGB-D images that are semantically segmented using a neural network (Section 2.2.3) before being projected into the mapping frame as a point cloud $P_M \subset (\mathbb{R}^{3+k})^n$ (Lines 5-7, Alg. 1) made up of n points (Section 2.2.2). The first three components of a point $\boldsymbol{p}_M \in P_M$ correspond to the Euclidean coordinates of the point in the mapping frame, while the last k components correspond to the terrain class probability score output from the semantic segmentation network. Next, we project points $\boldsymbol{p}_M \in P_M$ and the mesh vertex coordinates $\boldsymbol{v} \in \mathcal{V}$ onto the ground plane $\mathcal{G}$ by projecting $\boldsymbol{p}_M$ and $\boldsymbol{v}$ onto their first two coordinates. We obtain the projected point $\tilde{\boldsymbol{p}}_M = [p_x, p_y]$ as well as the three projected vertices $\tilde{\boldsymbol{v}}_i = [v_{xi}, v_{yi}]$ of a mesh element ξ for each $i \in \{1, 2, 3\}$. Given $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{v}}$, we compute the Barycentric coordinates

Algorithm 2: Assign points $\boldsymbol{p}_M \in P_M$ to $\xi \in \Xi$

Requires : P_M, $\mathcal{V}$, Ξ

1 **for** $\boldsymbol{p}_M \in P_M$ **do**
2 **for** $\xi \in \Xi$ **do**
3 $\tilde{\boldsymbol{p}}_M, \tilde{\boldsymbol{v}} \leftarrow \text{groundPlaneProjection}(\xi, \boldsymbol{p}_M)$
4 $\boldsymbol{\lambda} \leftarrow \text{computeBarycentricCoords}(\tilde{\boldsymbol{p}}_M, \tilde{\boldsymbol{v}})$
5 **if** for all $\lambda_i \in \boldsymbol{\lambda}, \lambda_i \in [0, 1]$ **then**
6 $\xi \leftarrow \text{addInteriorPoint}(\boldsymbol{p}_M)$

$\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]$ using the following linear transform (Line 4, Alg. 2):

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ v_{x1} & v_{x2} & v_{x3} \\ v_{y1} & v_{y2} & v_{y3} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ p_x \\ p_y \end{bmatrix}. \tag{2.8}$$

We apply Theorem 1 to determine whether to assign a point as an interior point to mesh element ξ (Line 6, Alg. 2).

2.4.3 Elevation Map Computation

Next we describe how Algorithm 3 recursively estimates the elevation map given the range sensor measurements (Line 9, Alg. 1). These interior points from the preceding section are now used to update the elevation map.

For a vertex $\boldsymbol{v} \in \mathcal{V}$, we take the interior points from the surrounding mesh elements (Line 5, Alg. 3) and apply a 1-dimensional Kalman filter update to estimate the mean height v_z and variance v_{σ_z} of the vertex. Given that the depth image used to compute the point cloud P_M has sensor noise, there is variance in the Euclidean coordinates of $\boldsymbol{p}_M \in P_M$. Once assigned to a mesh element, the elevation map depends only on the height of the points $\boldsymbol{p}_M \in P_M$, so we only consider the variance of the third Euclidean coordinate.

Recall that the third component of $\boldsymbol{p}_M$, which we denote $p_{M,3}$ describes its height. By the error propagation law [168], the variance of $p_{M,3}$ is computed (Line 7, Alg. 3):

$$\sigma^2 = J_s \Sigma_s J_s^\intercal + J_p \Sigma_p J_p^\intercal, \tag{2.9}$$

where Σ_s and Σ_p are the range sensor measurement noise and the sensor pose covariance matrix, respectively, and J_s and J_p, are constructed by taking the following partial deriva-

Algorithm 3: Update Elevation Map

Requires : $\mathcal{V}$, Ξ

1 $\boldsymbol{q} \leftarrow$ cameraPose()
2 $\Sigma_s \leftarrow$ sensorNoiseModel()
3 $\Sigma_p \leftarrow$ robotPoseCovariance()
4 **for** $\boldsymbol{v} \in V$ **do**
5 $\overline{P}_M \leftarrow$ getSurroundingInteriorPoints($\boldsymbol{v}, \Xi$)
6 **for** $\overline{\boldsymbol{p}}_M \in \overline{P}_M$ **do**
7 $\overline{p}_{M,3}, \sigma^2 \leftarrow$ heightVariance($\overline{\boldsymbol{p}}_M, \boldsymbol{q}, \Sigma_s, \Sigma_p$)
8 $v_z, v_{\sigma^2} \leftarrow$ 1DKalmanFilter($\overline{p}_{M,3}, \sigma^2$)

Return : $\mathcal{V}, \Xi$

tives:

$$J_s := \frac{\partial p_{M,3}}{\partial \boldsymbol{p}_{\mathcal{S}}} = (R_{\mathcal{S}}^{\mathcal{M}\intercal}(\boldsymbol{q}))_3 \tag{2.10}$$

$$J_p := \frac{\partial p_{M,3}}{\partial R_{\mathcal{S}}^{\mathcal{M}}(\boldsymbol{q})} = (R_{\mathcal{S}}^{\mathcal{M}\intercal}(\boldsymbol{q}))_3 \times \boldsymbol{p}_{\mathcal{S}}, \tag{2.11}$$

where $(R_{\mathcal{S}}^{\mathcal{M}\intercal}(\boldsymbol{q}))_3$ denotes the third row of $R_{\mathcal{S}}^{\mathcal{M}\intercal}(\boldsymbol{q})$ and $\times$ denotes the cross product. The mean and variance of the vertex height, v_z and v_{σ^2}, are updated using a 1-dimensional Kalman filter (Line 8, Alg. 3) for all the interior points from the surrounding mesh elements:

$$v_z \leftarrow \frac{v_z \cdot \sigma^2 + z \cdot v_{\sigma^2}}{\sigma^2 + v_{\sigma^2}} \tag{2.12}$$

$$v_{\sigma^2} \leftarrow \frac{v_{\sigma^2} \cdot \sigma^2}{v_{\sigma^2} + \sigma^2}. \tag{2.13}$$

2.5 Recursive Terrain Property Estimation

The objective of our semantic mapping algorithm is to estimate the distribution of terrain properties of the environment around the robot. Motivated by prior work [129, 5], we use data to construct a conditional probability distribution, $f(\boldsymbol{\psi} \mid z = i)$, of terrain property, $\boldsymbol{\psi}$, conditioned on a terrain class estimate for a region $z = i$. Using this model, given $\mathcal{Z}$ measurements of a region that is interior to ξ, we then apply the Law of Total Probability to compute this region's predicted terrain property:

$$f(\boldsymbol{\psi} \mid \mathcal{Z}, \boldsymbol{\alpha}) = \sum_{i=1}^{k} f(\boldsymbol{\psi} \mid z = i) f(z = i \mid \mathcal{Z}, \boldsymbol{\alpha}) \tag{2.14}$$

Note, this chapter is interested in estimating the friction coefficient; however, the presented theory can be extended to other terrain properties of interest. We next discuss the components of (2.14) before presenting a closed-form solution for recursively estimating the coefficient of friction within our semantic map (Line 10, Alg. 1).

Following Section 2.4.2, semantically segmented pixels $\mathcal{Z}$ are projected into the mapping frame and assigned as interior points to mesh elements ξ. Recall from Sections 2.2.3, 2.2.4 and (2.2), the pixel-wise terrain class probability generated from a semantic segmentation network represent parameters $\boldsymbol{\theta}$, which are used to update $\boldsymbol{\alpha}$ via (2.6). For each mesh element, we compute $f(z = i \mid \mathcal{Z}, \boldsymbol{\alpha})$ using (2.7).

Terrain properties are not constant across a terrain class and thus should not be estimated by a single value. Rather, these properties should be modelled using a conditional probability distribution $f(\boldsymbol{\psi} \mid z = i)$. This model is fit using data collected from each class. As we show in Section 2.6, we create a well-fit model by selecting an appropriate mean μ_i and variance σ_i^2 for a unimodal Gaussian distribution $f(\boldsymbol{\psi} \mid z = i) = \mathcal{N}(\mu_i, \sigma_i^2)$.

Substituting (2.7) and the formula for the unimodal Gaussian into (2.14) gives a closed-form estimate for the terrain properties within a mesh element:

$$f(\boldsymbol{\psi} \mid \mathcal{Z}, \boldsymbol{\alpha}) = \sum_{i=1}^{k} \frac{\alpha_i}{\sum_{j=1}^{k} \alpha_j} \mathcal{N}(\mu_i, \sigma_i^2). \tag{2.15}$$

This is a multimodal Gaussian distribution where each mode is weighted relative to the recursively updated terrain class likelihood. Note that (2.15) can be extended to use terrain property models other than the unimodal Gaussian distribution.

2.6 Implementation

This section describes the implementation of our algorithm.

Algorithm 1 is implemented in C++ and includes a Robot Operating System (ROS) interface[1]. Our implementation features noise models for the Realsense RGB-D camera and an interface to include additional sensor noise models. We evaluated our method on a desktop with a 3.1GHz Ryzen 3600 processor, 32GB of RAM and an Nvidia RTX 2080 Ti GPU.

We use (2.14) to estimate terrain properties from semantically segmented RGB-D images. This requires a model relating terrain class to terrain properties. The dataset published in [137] is insufficient to compute a probabilistic model as it only contains approximately three friction measurements per terrain class, and neither the Gaussian friction model proposed

[1] https://github.com/roahmlab/sel_map

in [132] nor their friction data is currently publicly available. To compute a probabilistic model we introduce a novel dataset of friction measurements across ten terrain classes and make this data publicly accessible.[2] We discuss the steps for data collection and subsequent model fitting in the following paragraphs.

Our primary focus in this chapter is on estimating the coefficient of friction. We built a device to measure the coefficient of friction using the pulling force measured using a load cell, the known weight of the device, and $g = 9.81\frac{m}{s^2}$:

$$\mu = \frac{F_{pull}}{mg}. \tag{2.16}$$

Approximately ten thousand data samples were collected and the data was post-processed using a low-pass filter to remove measurement noise from the load cell. To model $f(\boldsymbol{\psi} \mid z = i)$, we fit the unimodal Gaussian, Weibull, and log-normal distributions, and we assessed the goodness-of-fit for each distribution using the Kolmogorov-Smirnov test [113]. The unimodal Gaussian distribution had the highest average Kolmogorov-Smirnov score across all terrain classes demonstrating that the Gaussian model generalized the best over the entire dataset. We therefore use the unimodal Gaussian model to model $f(\boldsymbol{\psi} \mid z = i)$. Table 2.1 contains the mean and variance parameters of each unimodal Gaussian distribution for each terrain class of interest.

2.7 Results

We evaluate the performance of our mapping framework in the Carla simulation environment [42] and on a physical legged robot. In simulation, we compare our method against two baselines representing state-of-the-art terrain property estimation methods and illustrate that our method outperforms both baselines. We also demonstrate our method in real-world indoor and outdoor environments on a quadruped robot and compare it to a state-of-the-art traversability estimation method. A supplementary video [3] demonstrates the proposed mapping framework on the Spot quadruped.

2.7.1 Computational Performance Evaluation

We ran Alg. 1 using a 1m×1m mesh and varied mesh element lengths with random input images and associated ground-truth semantic segmentations to evaluate the computational

[2] https://github.com/roahmlab/terrain_friction_dataset
[3] https://roahmlab.github.io/sel_map/

Table 2.1: Unimodal Gaussian parameters computed from coefficient of friction data collected across multiple terrain classes.

Semantic Class	Gaussian Parameters	
	μ	σ
Concrete	0.543	0.065
Grass	0.577	0.077
Rock	0.478	0.133
Pebbles	0.428	0.059
Wood	0.372	0.055
Rubber	0.616	0.048
Plastic	0.311	0.045
Rug	0.583	0.068
Snow	0.390	0.071
Ice	0.192	0.046

speed and memory requirements. Approximately 45-55MB of memory is required to store the mesh. With 1cm mesh element lengths, Algorithm 1 takes 527ms to run, of which the semantic segmentation network from [214] takes 477ms (Alg. 1 Lines 6-7), and the elevation map and terrain property update takes 50ms (Alg. 1 Lines 8-10). These computation times were computed by averaging across 300 trials. The computation time for semantic segmentation is network dependent, and using Fast-SCNN [145] the total computation time is reduced to approximately 200ms. A thorough evaluation of the computational times for Algorithm 1 with two semantic segmentation networks, [214] and [145], and varying mesh element lengths is given in Figure 2.4.

2.7.2 Simulation

We evaluate our terrain property estimation method in the Carla simulation environment [42] where ground truth terrain property information is provided on a per-class basis. The ground truth distribution for the coefficient of friction for each class is a unimodal Gaussian using the coefficient of friction models computed in Section 2.6. Within Carla, we collect RGB-D information from a camera mounted on the front of a car. To estimate terrain class, we use the pre-trained semantic segmentation network presented in [214] and trained on the ADE20K dataset.

We use coefficient of friction estimates to compare our method against two baselines representing the state of the art in the terrain property estimation literature. The first baseline, denoted as the *Unimodal Non-Recursive* method, estimates the coefficient of friction by taking the most likely terrain class for a given mesh element at each instance in time and uses the

Table 2.2: The Kullback-Leibler divergence scores, average precision, and average accuracy of the two baselines and our method when applied to the Carla simulation environment. Arrows depict whether a high (↑) or low (↓) score is desired. A bolded score indicates the best performing method in each criteria.

Method	Kullback-Leibler Score (↓)	Average Precision (↑)	Average Accuracy (↑)
Uni-Modal Non-Recursive	43.3	0.59	0.58
Multi-Modal Non-Recursive	3.7	**0.99**	0.93
Ours	**2.4**	**0.99**	**0.95**

unimodal Gaussian model with parameters from Table 2.1. This baseline is representative of methods such as those presented in [194, 19, 147], which estimate the expected value of the coefficient of friction using convolutional neural networks. The second baseline, denoted as the *Multimodal Non-Recursive* method, uses the full categorical distribution of a given semantically segmented mesh element to estimate the coefficient of friction as a multi-modal Gaussian distribution. This equates to using (2.15) to compute the coefficient of friction directly from the pixel-wise categorical scores outputted from the semantic segmentation network. This baseline is representative of the state of the art methods [5, 132] that use the terrain class to estimate terrain properties from RGB-D images. These methods do not employ a recursive framework to update belief in terrain classifications. We ran our algorithm and the baselines offline using the data collected within Carla and compared these estimates with the ground-truth distributions using a precision-recall curve (Fig. 2.3) and their Kullback-Leibler divergence scores (Table 2.2).

Note, the lower the Kullback-Leibler divergence score, the more similar two distributions are. From Table 2.2, one can see that the first baseline performed poorly for the Carla dataset. The score for the second baseline is lower than the first's, indicating the coefficient of friction distribution estimate of the second baseline is more similar to the ground-truth distribution. Lastly, our proposed method performed best on the Kullback-Leibler divergence test and demonstrates that the coefficient of friction distribution estimated using our proposed method is the most similar to the ground-truth distribution.

The precision-recall curve summarizes the trade-off between the true positive rate and the positive predicted value and is used to evaluate the performance of a multi-class classifier. We use the average precision to evaluate the performance on the precision-recall curve as seen in Figure 2.3. A higher average precision indicates a more accurate classifier. For this evaluation, we divide the range of coefficient of friction values into low friction ($\mu \leq 0.5$) and high friction ($\mu > 0.5$) categories and compare the ability of our method and and the

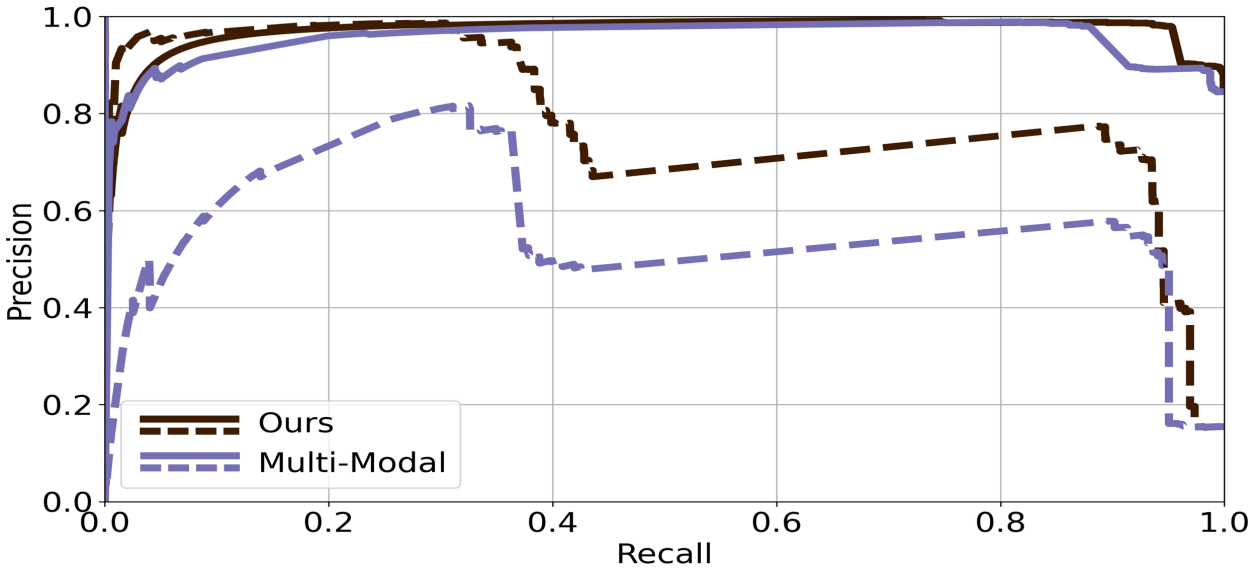


Figure 2.3: The Precision-Recall curve for terrain property estimation within the Carla simulator. We compare our method against the *Multimodal Non-Recursive* baseline for regions of high friction coefficients ($\mu > 0.5$), plotted using a solid line, and low friction coefficients ($\mu \leq 0.5$), plotted using a dashed line. Our method's performance is comparable to the baseline for regions of high friction, however, for regions of low friction our method significantly outperforms the baseline.

Multimodal Non-Recursive method to correctly predict whether a given mesh element falls within the low or high friction category. The results for *Unimodal Non-Recursive* method is omitted due to poor performance.

Table 2.2 includes the performance of all methods using the three quantitative metrics. Due to class imbalance within the simulation environment, more high-friction terrain classes are present in the data. The baselines perform better for high-friction classes, but even with this class imbalance our method matches or outperforms both baselines across all evaluation criteria. This shows our method is able to better predict the terrain friction properties than previous terrain estimation methods from the literature.

2.7.3 Real-World

We ran our method on the Spot quadruped using an on-board Realsense D435 RGB-D camera. Experiments were conducted both indoors and outdoors with a variety of terrain classes. We compared our method to a state-of-the-art traversability mapping framework [60] to demonstrate the utility of our semantic mapping algorithm when compared to a traversability estimation algorithm. Figure 2.5 illustrates the performance of both algorithms on a variety of examples across different terrains. On an icy surface (Fig. 5a), for instance,

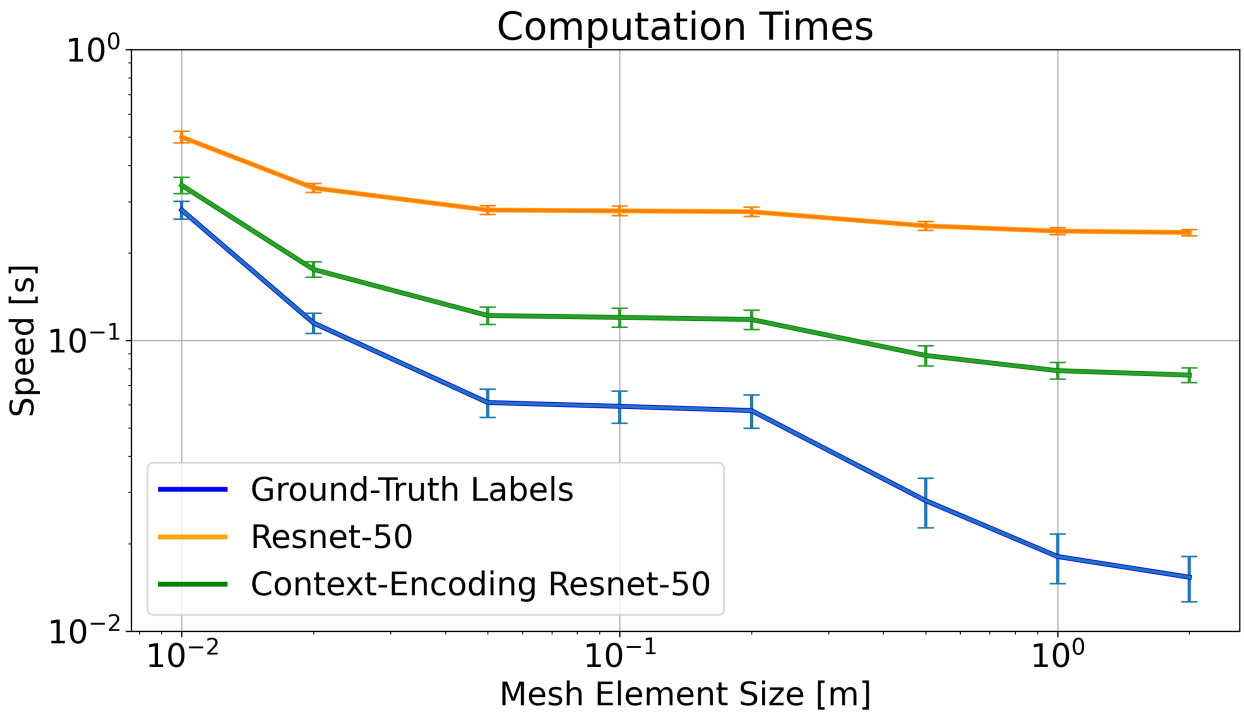


Figure 2.4: Computation times of Algorithm 1 for a 10m ×10m mesh with varying mesh element lengths using two off-the-shelf semantic segmentation networks, Resnet-50 [214] and Context-Encoding Resnet-50 [212], as well as the baseline algorithm assuming ground-truth semantically segmented images. The ground-truth label experiments use a pre-generated semantic segmentation image, representing the speed of Algorithm 1 without considering the time required for semantic segmentation (Line 6) and represents an lower bound on the speed of our algorithm. Error bars represent one standard deviation.

our method is able to predict the low friction of the surface, while the traversability estimate assumes the surface is safe to walk on and provides no additional information regarding the surface. Similarly, Figs. 5b and 5c illustrate that the traversability estimation incorrectly classifies regions which are traversable as intraversable while our method is able to predict the terrain geometry and properties. When no terrain class from Table 2.1 is estimated within a mesh element, we make no friction estimate and color the mesh element grey (second row, Fig. 2.5). Traversability depends on the means of robot locomotion and other robot-dependent factors. In an effort to generalize, traversability estimation methods often over- or under-approximate traversable regions, supporting the conclusions reached by [99]. In contrast, our method provides more information than binary traversability labels which better informs robots about their environment.

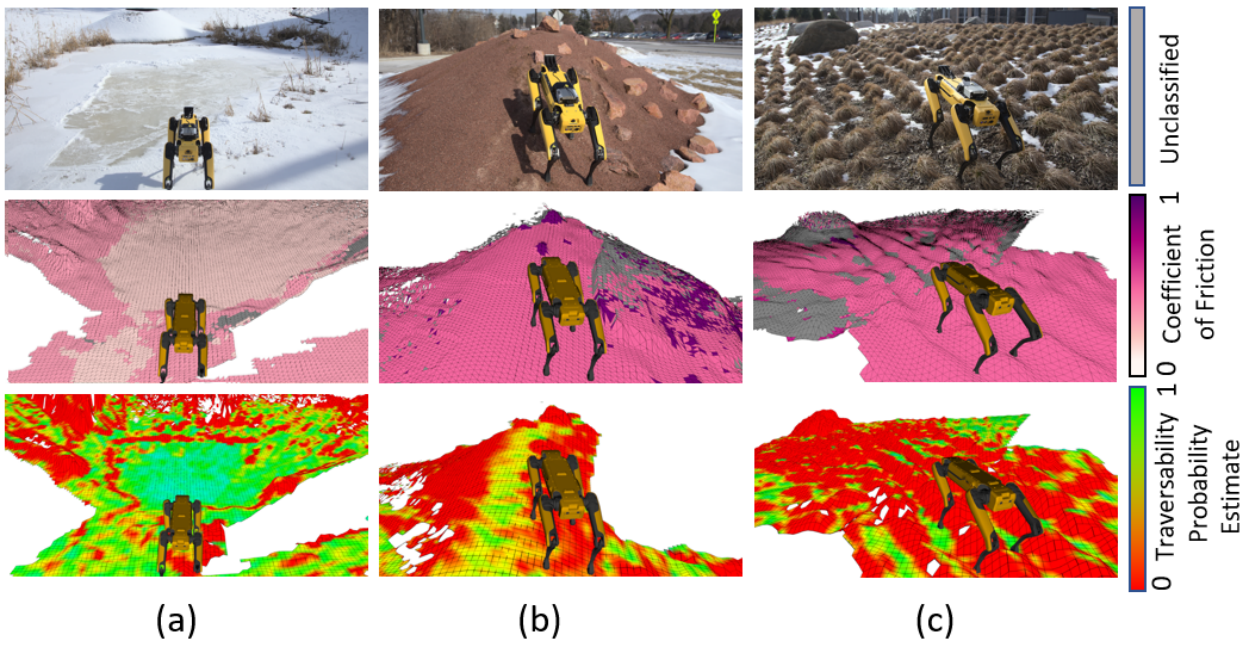


Figure 2.5: Each column depicts the performance of our proposed mapping algorithm (second row, the first column uses the Context-Encoding ResNet-50 trained on the Pascal dataset while the remaining columns use the RenNet-50 trained on ADE20K for terrain classification) when compared to a traversability estimation algorithm [60] (third row) applied on the scenes depicted in the top row. Traversability estimation sometimes believes that a region is traversable when it is not, such as an icy surface (Fig. 5a). In other scenarios, it believes that an area is intraversable when it is traversable such as on hills (Figs. 5b) and near low vegetation (Fig. 5c). Our method makes no claims about traversability, instead it estimates the probability distribution of terrain properties for each mesh element along with the terrain geometry.

2.8 Extensions to Voxel Mapping

The semantic classification uncertainty update derived in Section 2.2 may also be applied to voxel maps in [198], where each voxel stored the Dirichlet parameters for classification. In this formulation, the semantic update for each class can be computed via lidar point clouds with semantics computed via a neural network. In voxel maps, it is been proposed that neighboring voxels affect semantic classifications, and kernel-based methods have been used to compute the associated joint probability. We pose this kernel-based semantic mapping problem as a differentiable neural network layer (ConvBKI) in order to achieve the accelerated inference rates and optimized performance of deep learning while maintaining the ability to quantify uncertainty in closed form. The semantic Bayesian update from (2.6) is at the heart of this layer.

2.8.1 Semantic Prediction as a Convolution

For supervised learning problems, our goal is to identify the relationship $p(y_*|x*, \mathcal{D})$ from a sequence of N independent observations $\mathcal{D} = \{(x_1, y_1), ..., (x_N, y_N)\}$. In semantic mapping, the input data is a series of semantically labeled points with positions x_i and semantic observations y_i. Observations are associated with latent model parameters θ_i which define the likelihood $p(y_i|\theta_i)$. In the case of semantic mapping, we wish to infer the categorical parameters θ_i, such that the probability of observing a one-hot encoded semantic category y_i may be written as:

$$p(y_i|\theta_i) = \prod_{c=1}^{C} (\theta_i^c)^{y_i^c}. \tag{2.17}$$

As we observe 3D semantic point-pairs (x_i, y_i), our goal is to learn the latent target parameters θ_* which define the Categorical posterior of each voxel $*$ with centroid x_*. Note that the voxel posterior $p(\theta_*|x_*, \mathcal{D})$ is dependent on observations continuously scattered around the voxel, requiring an update which considers the relative position of input data.

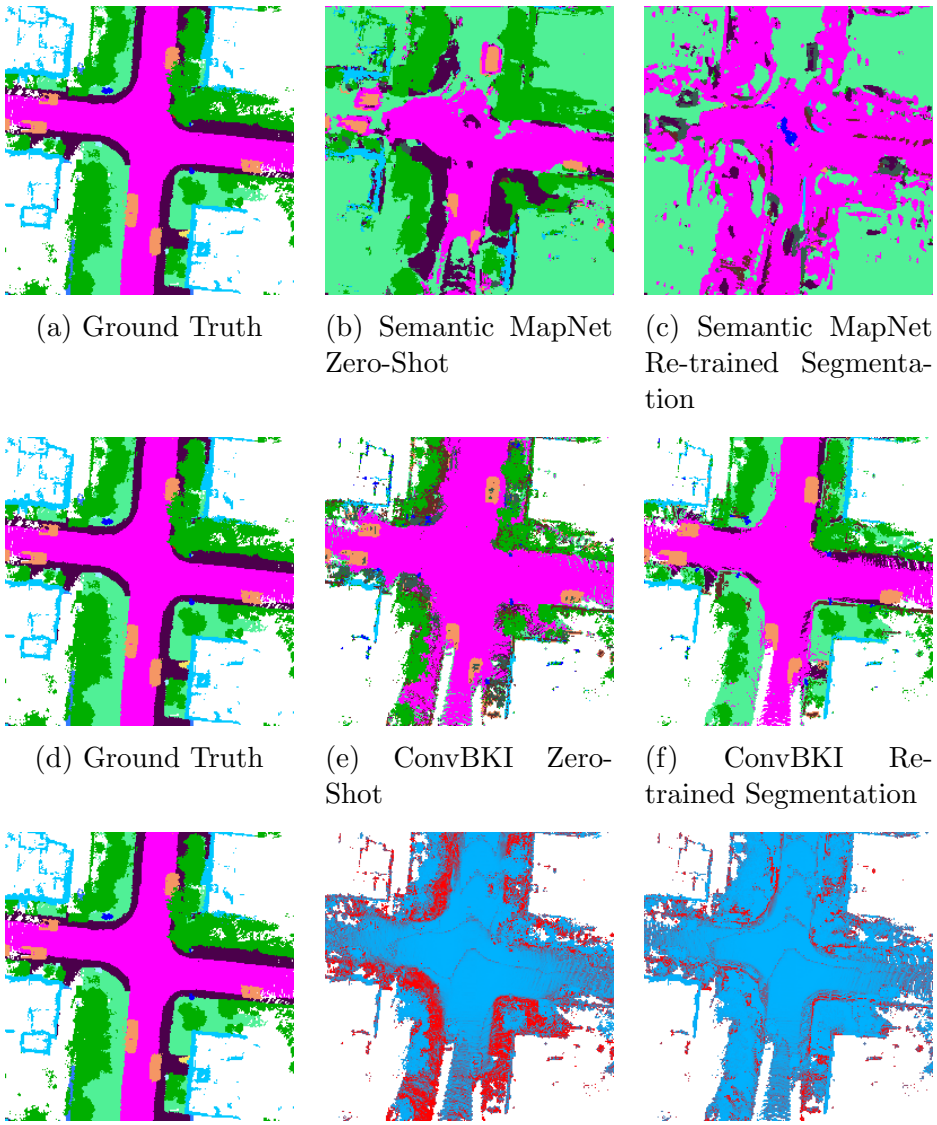
Vega-Brown et al. [190] propose a solution to relate the extended likelihood $g(y_i) = p(y_i|\theta_*, x_i, x_*)$ to the likelihood distribution $f(y_i) = p(y_i|\theta_*)$ through a positional kernel $k(x_i, x_*)$. This relationship generalizes local kernel estimation to Bayesian inference for supervised learning. In particular, they relate the extended likelihood and likelihood by showing that the maximum entropy distribution g satisfying $D_{KL}(g||f) \geq \rho(x_*, x_i)$ has the form $g(y_i) \propto f(y_i)^{k(x_*, x_i)}$ with kernel function k. In this case, $\rho : X \times X \to \mathbb{R}^+$ is some function which bounds information divergence between the likelihood distribution $f(y_i) = p(y_i|\theta_*)$ and the extended likelihood distribution $g(y_i) = p(y_i|\theta_*, x_i, x_*)$. Concretely,

$$p(y_i|x_i, \theta_*, x_*) \propto p(y_i|\theta_*)^{k(x_i, x_*)}. \tag{2.18}$$

Functions k and ρ have an equivalence relationship, where each is uniquely determined by the other. The only requirements are that

$$k(x, x) = 1 \forall x \quad \text{and} \quad k(x, x_*) \in [0, 1] \forall x, x_*. \tag{2.19}$$

As described above, in semantic mapping θ_* describes a Categorical distribution over the semantic categories C. Therefore, the extended likelihood spatially relates semantic observations (x_i, y_i) to voxel latent parameters θ_* using a kernel function k. This formulation is especially useful for likelihoods $p(y_i|\theta_*)$ chosen from the exponential family, as the likelihood raised to the power of $k(x_*, x)$ is still within the exponential family.



(a) Ground Truth

(b) Semantic MapNet Zero-Shot

(c) Semantic MapNet Re-trained Segmentation

(d) Ground Truth

(e) ConvBKI Zero-Shot

(f) ConvBKI Re-trained Segmentation

(g) Ground Truth

(h) ConvBKI Variance Zero-Sho

(i) ConvBKI Variance Re-trained Segmentation

Figure 2.6: Comparison of ConvBKI and Semantic MapNet on semantic mapping transferring from simulated to real data. Regions in white are unobserved by ConvBKI or in the ground truth. ConvBKI is generally successful, however suffers issues on ground segmentation due to LiDAR input. Ground segmentation is also a challenge on the simulated data, as the input semantic segmentation network is sometimes unable to distinguish sidewalk from road.

Doherty et al. [40] then apply the BKI kernel model to the task of occupancy mapping. In occupancy mapping, occupied points are measured by a 3D sensor such as LiDAR, and free space samples can be approximated through ray tracing. Measurement $x_i \in \mathbb{R}^3$ represents

a 3D position with corresponding observation $y_i^c \in \{0, 1\}$ where $(y_i^1 = 1)$ for free space measurements and $(y_i^2 = 1)$ for occupied measurements at the end of a ray. The voxel occupancy can be modeled by a Bernoulli distribution over θ_*. Adopting the conjugate prior distribution $\text{Beta}(\alpha_*^1, \alpha_*^2)$ over voxel parameters θ_* yields a closed-form update equation by applying Bayes' rule:

$$p(\theta_*|x_*, \mathcal{D}) \propto p(\mathcal{D}|\theta_*, x_*)p(\theta_*|x_*), \tag{2.20}$$

which can be written in terms of the extended likelihood as

$$p(\theta_*|x_*, \mathcal{D}) \propto \left[\prod_{i=1}^{N} p(y_i|x_*, x_i, \theta_*)\right] p(\theta_*|x_*). \tag{2.21}$$

Applying the relationship between extended likelihood and likelihood defined in (2.18), the posterior can be simplified to

$$p(\theta_*|x_*, \mathcal{D}) \propto \left[\prod_{i=1}^{N} p(y_i|\theta_*)^{k(x_i, x_*)}\right] p(\theta_*|x_*). \tag{2.22}$$

Due to the conjugate relationship between the prior $p(\theta_*|x_*)$ and likelihood $p(y_i|\theta_*)$ defined in (2.17), the update yields a closed-form solution where:

$$p(\theta_*|x_*, \mathcal{D}) \propto \left[\prod_{i=1}^{N} \left[\prod_{c=1}^{C} (\theta_*^c)^{y_i^c}\right]^{k(x_i, x_*)}\right] \prod_{c=1}^{C} (\theta_*^c)^{\alpha_*^c - 1}, \tag{2.23}$$

which simplifies to

$$p(\theta_*|x_*, \mathcal{D}) \propto \prod_{c=1}^{C} (\theta_*^c)^{\alpha_*^c + \sum_{i=1}^{N} y_i^c k(x_i, x_*) - 1}. \tag{2.24}$$

Since the posterior is proportional to the conjugate prior distribution $\text{Beta}(\alpha_*)$, the update at time-step t for voxel $*$ can be written in closed-form as

$$\alpha_{*,t}^c = \alpha_{*,t-1}^c + \sum_{i=1}^{N_t} k(x_*, x_i) y_i^c. \tag{2.25}$$

The equation provides a closed-form method for updating the belief that voxel * is occupied or free given observed measurements and samples of free space.The un-normalized posterior $\text{Beta}(\alpha_{*,t}^1, \alpha_{*,t}^2)$ is a weighted summation of nearby inputs points, where influence of nearby points is calculated through the kernel function. From the un-normalized voxel

concentration parameters α_*, the normalized expectation and variance of voxel parameters θ_* are defined as

$$\eta_* = \sum_{j=1}^{C} \alpha_*^j, \quad \mathbb{E}[\theta_*^c] = \frac{\alpha_*^c}{\eta_*}, \quad \mathbb{V}[\theta_*^c] = \frac{\frac{\alpha_*^c}{\eta_*}(1 - \frac{\alpha_*^c}{\eta_*})}{1 + \eta_*}. \tag{2.26}$$

Doherty et al. [40] propose to use a sparse kernel function [117] which simplifies kernel calculation to a function of the Euclidean distance between two points. For two points x_i and x_*, the sparse kernel is calculated as

$$\kappa(d) = \begin{cases} \sigma_0[\frac{1}{3}(2 + \cos(2\pi\frac{d}{l})(1 - \frac{d}{l}) + \frac{1}{2\pi}\sin(2\pi\frac{d}{l}))], & \text{if } d < l \\ 0, & \text{else} \end{cases} \tag{2.27}$$

where $d := \|x_i - x_*\|_2$ and l is the kernel length. The variables σ_0 is used to scale the magnitude of the kernel, however $\sigma_0 = 1$ to satisfy (2.19). Note that whereas the kernel function $k : \mathbb{R}^3 \times \mathbb{R}^3 \to [0, 1]$ is defined between two points, the sparse kernel function $\kappa : \mathbb{R}^3 \to [0, 1]$ is a function of the Euclidean distance between the two points.

Gan et al. [62] show that the same approach can be applied to semantic labels by adopting a Categorical likelihood over θ_* with conjugate prior $\text{Dir}(C, \alpha_0)$. This model is also calculated using (2.25), however the number of classes has increased from two to any positive integer C. Similarly, the free space class may be treated as another semantic category with samples also obtained by linearly interpolating along each ray. Semantic observations y_i are the one-hot encoded predictions from neural networks, yielding the likelihood described in (2.17).

Although Semantic BKI has achieved accurate and efficient performance in 3D mapping compared to previous methods, it is still limited in key ways. Firstly, the kernel is hand-crafted, and kernel parameters must be manually tuned. As a result, a single spherical kernel is shared between all semantic classes. Second, the update operation is computationally expensive, as the kernel inference must locate all nearby voxels and does not leverage the parallelization of modern GPU architectures.

First, we compare our network to its most direct probabilistic comparisons on the KITTI Odometry dataset [65]with semantic labels obtained from [162] following the approach of Semantic BKI [62]. The purpose of comparing ConvBKI against probabilistic baselines is to highlight the quantitative advantages of kernel optimization, as well as the accelerated inference rate.

The test set contains a sequence of one hundred consecutive raw RGB images with depth estimated by ELAS [66] and pose estimated by ORB-SLAM [127]. Semantic segmentation predictions for the image were generated by the deep network dilated CNN [209]. Ground

Table 2.3: Results on KITTI Odometry sequence 15 [162]. As the degree of kernel expressivity increases, mIoU of ConvBKI also improves from 76.5% without optimization to 77.7% with fully optimized per-class compound kernels.

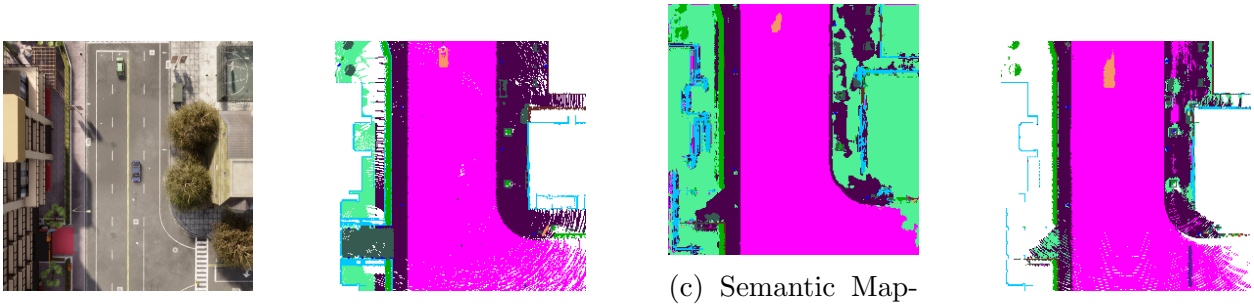
Method	Building	Road	Vege.	Sidewalk	Car	Sign	Fence	Pole	**mIoU (%)**	**Freq. (Hz)**
Segmentation [209]	92.1	93.9	90.7	81.9	94.6	19.8	78.9	49.3	75.1	
Yang et al. [207]	**95.6**	90.4	**92.8**	70.0	94.4	0.1	**84.5**	49.5	72.2	
BGKOctoMap-CRF [40]	94.7	93.8	90.2	81.1	92.9	0.0	78.0	49.7	72.5	
S-CSM [62]	94.4	95.4	90.7	84.5	95.0	22.2	79.3	51.6	76.6	
S-BKI (fine)	94.6	95.4	90.4	84.2	95.1	**27.1**	79.3	51.3	77.2	0.6
S-BKI (0.2m)	92.6	94.7	90.9	84.5	95.1	21.9	80.0	52.0	76.5	
ConvBKI Single	92.7	94.8	90.9	84.7	95.1	22.1	80.2	52.1	76.6	
ConvBKI Per Class	94.0	95.5	91.0	87.0	95.1	22.8	81.8	52.9	77.5	
ConvBKI Compound	94.0	**95.6**	91.0	**87.2**	**95.1**	22.8	81.9	**54.3**	**77.7**	**44.3**

truth semantic segmentation labels are available for twenty-five of the one hundred input images test set images from [162]. Each baseline recurrently generates a global map from the stream of time-synchronized pose, depth, and semantic segmentation predictions. The goal of each algorithm is to estimate semantic segmentation labels at the current time step, given all past information. Therefore, baselines are compared by projecting pixels with depth to 3D world coordinates and assigning semantic segmentation to each pixel corresponding to the voxel the 3D projection resides in. Semantic predictions are aggregated over all time-steps in a global semantic map, and mapping accuracy is evaluated by the Intersection over Union (IoU) metric. The IoU metric for the j-th semantic class is computed as

$$IoU_j = \frac{\max(\mathbb{E}[\theta_j]) \bigcap z_j}{\max(\mathbb{E}[\theta_j]) \bigcup z_j} \tag{2.28}$$

where z_j are the ground-truth voxels of the j-th class. The mean IoU (mIoU) is then the mean over all the semantic classes within the dataset.

Baselines include a CRF-based semantic mapping system [207], BGKOctoMap-CRF [40, 62], S-BKI [62] and Semantic-Counting Sensor Model (S-CSM) [62]. Our most direct comparison is S-BKI, as the goal of our network is to accelerate and optimize the algorithm in a deep learning framework. However, S-BKI uses a resolution of 0.1 meters without downsampling or discretization of the input point clouds to improve results. In contrast, our method discretizes input point clouds into voxels to simplify the kernel calculation as a



(a) BEV Camera (b) BEV Semantics (c) Semantic MapNet (d) ConvBKI

Figure 2.7: Comparison of ConvBKI and Semantic MapNet on BEV mapping in CarlaSC. Both networks correctly segment the driveable surface and buildings. Compared with Semantic MapNet, ConvBKI does not make predictions on unseen regions, creating some sparsity in the map. ConvBKI correctly identifies the pedestrian in the bottom right, however leaves a trace as the pedestrian walks along the crosswalk. In contrast, Semantic MapNet omits the pedestrian to achieve a higher quantitative performance.

convolution. Therefore, to directly evaluate the benefit of optimization, we add a version of S-BKI with a resolution of 0.2 meters and discretization, which we label S-BKI (0.2 meters). Grid bounds for ConvBKI are set to a range of [-40, -40, -5.0] to [40, 40, 5.0] meters. Results are summarized in Table 2.3.

Kernel Optimization: The results of Table 2.3 demonstrate the utility of optimizing kernels, as subsequently more expressive kernels achieve higher mIoU. Starting from S-BKI (0.2m) which represents an unoptimized version of ConvBKI with a single filter shared between all classes, performance is slightly better than the segmentation input. Even without optimization, the BKI framework is capable of recurrently integrating information to improve semantic segmentation. With optimization over a single kernel shared by all classes, denoted ConvBKI Single, performance slightly increases from 76.5 to 76.6% mIoU. The small increase is due to only a single parameter being optimized, yielding an un-expressive kernel. When a kernel is optimized for each semantic category (ConvBKI Per Class), mIoU increases substantially from 76.6 to 77.5%. Finally, dividing the kernel for each semantic category into the product of a horizontal and vertical kernel (ConvBKI Compound) again increases performance, this time most notably for the pole class. Intuitively, points vertically adjacent to points on a pole are also likely to be a pole, whereas horizontally adjacent points are not.

Latency: Another notable comparison in Table 2.3 is between S-BKI (fine) and ConvBKI Compound. S-BKI (fine) uses a finer resolution of 0.1m without discretization of the input point cloud compared to ConvBKI Compound which has a resolution of 0.2 m and discretization of input points into voxelized coordinates. However, by leveraging differentiability, ConvBKI Compound is able to improve semantic mapping performance to surpass

Table 2.4: Comparison with learning-based memory on simulated data. Metrics are reported for the isolated recurrent networks, excluding the input segmentation network.

Method	GRU	LSTM	Linear	ConvBKI
Map Dimension	2	2	2	**3**
Memory (MB)	2765	2917	2707	**2497**
Parameters	1.96 M	2.04 M	1.74 M	**22**
mIoU (1 epoch)	24.0	25.3	24.7	**26.7**
mIoU (15 epoch)	**35.4**	30.9	32.6	N/A
Freq. (Hz)	111.0	107.4	116.8	**176.6**
Data	BEV	BEV	BEV	**Segmentation**

that of S-BKI. Additionally, ConvBKI achieves significant acceleration, with an inference frequency of 44.3 Hz compared to 0.6 Hz for S-BKI. Even with downsampling of the input points, which is used by S-BKI for real-world performance, S-BKI reports an inference rate of 2 Hz. In summary, ConvBKI achieves an accelerated inference rate and optimized performance compared to probabilistic baselines.

2.8.2 Neural Network Comparison

Next, we compare ConvBKI Compound against several types of memory for recurrent neural networks on the task of Bird's Eye View (BEV) semantic mapping. The purpose of this evaluation is to isolate the trade-offs of ConvBKI opposed to more expressive algorithms with thousands of times more parameters. Specifically, we compare the ground-truth data used, memory, quantitative performance, inference rate, and reliability when transferring to a different dataset.

In contrast to the global mapping comparison for probabilistic mapping, we evaluate on local BEV semantic mapping for a learning comparison. BEV mapping is more common for learning-based algorithms due to the memory cost of 3D operations, and provides a measure of scene understanding by evaluating the semantic map of ConvBKI from a different persepective.

We chose to compare with Semantic MapNet [25] as it was the most direct open-source learning-based comparison we were able to find. Semantic MapNet is a BEV semantic mapping algorithm which uses features from the last layer of a pre-trained semantic segmentation

algorithm as input to per-cell recurrent units which perform a map update in feature space. Semantic MapNet groups points with features from the semantic segmentation network into square cells and applies a BEV filter which discards all but the highest point in each cell. Each cell with an input point is then updated through a learned memory model, including a Gated Recurrent Unit (GRU) [32], Long Short-Term Memory (LSTM) [81] or a linear layer. At inference time, a series of convolution layers are applied to the latent-space memory to produce a BEV semantic segmentation prediction for each 2-dimensional cell.

Semantic MapNet functions similarly to ConvBKI except for several key differences. First, Semantic MapNet operates in two dimensions using only the highest point in each map cell as input, compared to the 3D representation of ConvBKI which incorporates every input point. Additionally, Semantic MapNet applies a recurrent network which updates each cell using latent features of the input points, whereas ConvBKI applies an explicit Bayesian update on the semantic predictions of the semantic segmentation network.

The input to each ConvBKI and Semantic MapNet is from the same semantic segmentation network, Sparse Point-Voxel Convolutional Neural Network (SPVCNN) [177]. SPVCNN is an efficient semantic segmentation for LiDAR point clouds which leverages sparsity to accelerate inference. We follow the training procedure from Semantic MapNet with two modifications, including changing the number of input frames to $\tau = 10$ to match ConvBKI and the batch size to 2 in order to reduce training memory. Semantic MapNet was trained for fifteen epochs over the course of three days. For evaluation, each network is provided the past $\tau = 10$ frames as input and evaluated on the accuracy of the BEV local semantic map produced. Following the evaluation procedure of Semantic MapNet, cells with no input are discarded from evaluation.

2.8.3 Simulation Results

First, we train ConvBKI and Semantic MapNet on the simulated driving dataset CarlaSC gathered from the CARLA simulator [200]. CarlaSC contains complete 3D scenes with ground truth semantic labels for each voxel at a resolution of 0.2 m and a sensor configuration designed to mimic the popular real-world Semantic KITTI [13] driving dataset. Additionally, CarlaSC is a highly dynamic dataset with moving actors which we use to highlight the artifacts left behind by ConvBKI.

Ground truth BEV semantic maps are obtained by ray-tracing within each ground truth 3D map until an occupied voxel is found, and assigning the BEV semantic label to match the first occupied voxel. If no occupied voxel is encountered, the 2D cell is ignored during training and evaluation. We compare ConvBKI and Semantic MapNet on the test set of

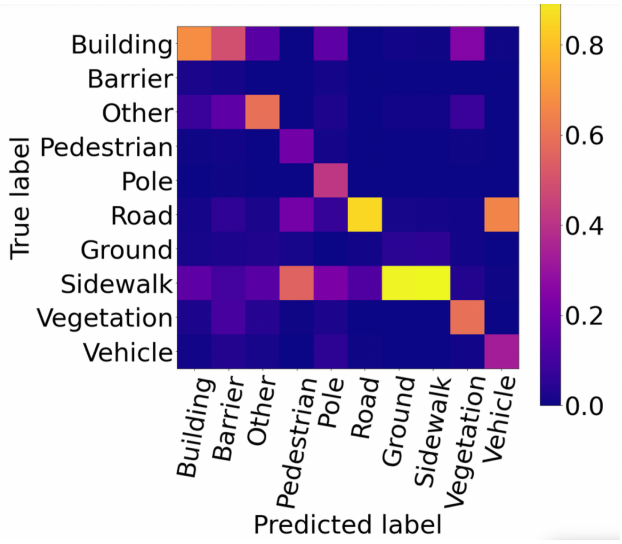


Figure 2.8: Confusion matrix of ConvBKI compound on simulated BEV mapping, normalized over the predictions. ConvBKI leaves traces from dynamic objects, causing sidewalk to be incorrectly labeled as pedestrian and road as vehicle.

Table 2.5: Per-class comparison of ConvBKI and SemanticMapNet (SMNet) on simulated data. Key results are bolded. Performance of both algorithms is similar, except on the dynamic object categories. ConvBKI leaves traces which penalizes precision of dynamic objects and recall of ground surfaces.

Method	Metric	Building	Barrier	Other	Pedestrian	Pole	Road	Ground	Sidewalk	Vege.	Vehicle
ConvBKI	**Prec.**	59.8	1.3	51.5	**10.5**	**29.3**	85.1	5.2	91.2	57.4	**21.3**
SMNet	**Prec.**	67.7	1.3	43.5	31.5	50.2	87.2	6.06	92.6	60.3	56.5
ConvBKI	**Rec.**	28.3	18.5	20.4	**72.4**	44.9	**88.7**	42.4	**41.3**	85.4	69.4
SMNet	**Rec.**	34.2	13.2	26.0	35.4	55.3	98.1	22.5	67.7	88.9	53.0
ConvBKI	**IoU**	23.8	1.2	17.1	10.1	21.5	76.8	4.9	39.7	52.3	19.4
SMNet	**IoU**	29.4	1.2	19.4	20.0	35.7	85.8	5.0	64.2	56.1	37.7

CarlaSC, which includes three scenes of 3 minutes in length sampled at 10 Hz. We use the remapped CarlaSC labels which contain 11 semantic categories: free space, building, barrier, other, pedestrian, pole, road, ground, sidewalk, vegetation and vehicles. Models are evaluated on the mIoU of all semantic predictions in the 5,400 frame test-set. Ground truth scenes are at a resolution of 0.2 meters with local boundaries with respect to the on-board LiDAR sensor of (-25.6, -25.6, -2.0) meters to (25.6, 25.6, 1.0) meters.

Efficiency: Key metrics on the test set of CarlaSC are summarized in Table 2.4. Com-

pared to the learning-based baselines, ConvBKI builds and maintains a *three dimensional* map using *less memory* than the baselines take to operate in two dimensions. This difference is because the ConvBKI update is explicitly defined to achieve optimal efficiency by building off prior probabilistic mapping works. ConvBKI only requires two parameters for each semantic category to learn the geometry, for a total of 22 parameters. In contrast, the learning-based memory models contain nearly a hundred thousand times more parameters due to purely implicit operations. While implicit operations potentially enable more expressive functions to be learned, they are also less reliable as demonstrated in the next section. Additionally, ConvBKI achieves a quicker inference rate despite operating in 3-D due to maximally efficient operations. Note that the inference rate is different from Table 2.3 due to the change in grid size.

Training: Another advantage of ConvBKI is the form of training data. ConvBKI trains on semantic segmentation ground-truth data which is more easily obtained than ground-truth BEV maps. Obtaining accurate ground-truth maps in the real world is a difficult challenge due to the presence of dynamic objects [200], requiring multiple viewpoints for accurate data. Semantic segmentation ground-truth data is more easily obtained, as evident by the vast number of on-road driving datasets.

The primary trade-off of ConvBKI is highlighted by the BEV semantic segmentation performance. ConvBKI trains quickly in less than an epoch, but is gradually outperformed by implicit memory models as they train over the course of several days. With nearly a hundred thousand times more parameters, Semantic MapNet is able to outperform ConvBKI on BEV segmentation. ConvBKI obtains a higher mIoU of 26.7% after one epoch of training compared to the implicit memory models, but is surpassed by the GRU after several days of training which obtains an mIoU of 35.4%.

Dynamic Objects: The lower mIoU of ConvBKI can be explained by examining the precision and recall of each semantic category in Table 2.5. ConvBKI is comparable to Semantic MapNet across most semantic categories, however is penalized on the dynamic classes due to the lack of a dynamic object propagation step considering object motion. ConvBKI recurrently incorporates semantically segmented points into the semantic map without separate treatment of dynamic objects, leaving traces. Measured on BEV mapping, traces cause lower precision of the pedestrian and vehicle classes, as well as lower recall of the sidewalk and road classes. Dynamic methods have already been proposed which fit within the BKI framework [185], however they do not fit into the scope of this chapter and we have left the integration of dynamic mapping with our accelerated network as future work.

A confusion matrix of ConvBKI is shown in Fig. 2.8, and a table of the precision, recall, and IoU per class compared to the fully trained Semantic MapNet is shown in Table 2.5. In

general, the performance between the two methods is very similar amongst all classes, with a few exceptions which motivates future work. Some notable columns include pedestrian and vehicle, where ConvBKI has a higher recall yet much lower precision due to artifacts from dynamic objects. In ConvBKI, dynamic objects are treated the same as static objects, which leave traces in the map. When projecting from 3D to 2D, the traces cause ground tiles to be incorrectly labeled as a dynamic object, thereby reducing precision. For the pedestrian class, many pedestrian labels are incorrectly applied to sidewalk tiles, and the same for vehicle predictions to the road class. Note that this is due to the highly dynamic nature of the CarlaSC simulated dataset, and would not be present in static scenes where performance would be higher. The same effect is seen on the road and sidewalk classes, where ConvBKI has a similar precision but lower recall since the BEV projection captures traces left behind by dynamic objects.

Reliability: Another key result is highlighted by the groups of ground classes and building/barrier classes. Our method is ultimately reliant on the input semantic predictions of the semantic segmentation network. Although ConvBKI may smooth out noise, it is incapable of resolving errors where the segmentation predicts temporally consistent incorrect labels. However, the predictable nature of our algorithm may also be seen as an advantage for improved reliability, as observed occupied voxels will always be modeled within our map, even if the label is wrong.

In Fig. 2.7, a pedestrian walks along the bottom right on a sidewalk. ConvBKI leaves a trace as it does not yet handle dynamic objects, while Semantic MapNet completely removes the pedestrian. This occurs because deep neural networks are constructed to optimize a specific metric, in this case mIoU, and may hallucinate to create the best performance at the cost of wrong predictions. In contrast, ConvBKI is heavily penalized on the pedestrian and road class for leaving traces. In safety-critical environments such as self-driving cars, however, it is vital to have a trustworthy system which does not collide with pedestrians or other difficult semantic categories, even if the semantic label is incorrect.

2.9 Extensions to Open-Vocabulary Mapping

The semantic classification uncertainty update derived in Section 2.2 applies to the so-called "closed-set" vocabulary, meaning there are a fixed number of predefined semantic classes. With the advent of large language models, vision-language models have become a popular means of classifying semantics using "open-set" vocabularies. These open-sets correspond to an $\mathbb{R}^n$ feature space, where the image's language embedding is defined as a point in this continuous space. In this formulation, the semantic update for each class may still be computed, however a different Bayesian update scheme must be used as the Dirichlet distribution defines a belief over a closed-set of semantic classes.

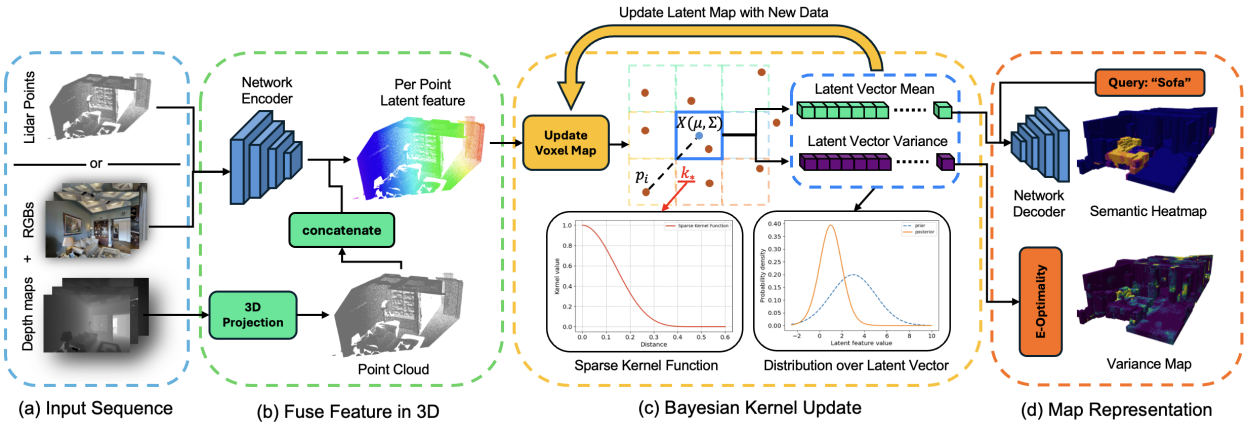


Figure 2.9: This demonstrates the overall pipeline of LatentBKI. (a) The input to LatentBKI is 3D points, which can be from LiDAR, RGB-D, or any exteroceptive sensor with 3D input. (b) Points are then processed by an off-the-shelf neural network, which encodes each point into a latent space. (c) By adopting a Gaussian likelihood over the point-wise features, we perform closed-form Bayesian inference on a voxel map where each voxel contains parameters modeling the conjugate prior of the multivariate Gaussian distribution. Additionally, instead of only considering points which fall within a voxel, we consider nearby points weighted through a kernel function. (d) The posterior predictive distribution of each voxel in latent space can then be decoded using the decoder of the neural network, enabling the computation of open-dictionary segmentation predictions with expectation and uncertainty.

In our recent work [201] we propose a novel method for probabilistic continuous mapping in the feature space of neural networks, which recurrently incorporates predictions from neural networks to learn an expectation and variance. Our mapping framework, which we call LatentBKI, has applications for general deep neural networks and is especially powerful when combined with modern foundation models such as VL models. Compared to previous methods which map in an explicit categorical space, continuous mapping in the feature space allows for open-dictionary queries with quantifiable uncertainty. A diagram of our method

is shown in Fig. 2.9, demonstrating the ability of LatentBKI to complete scenes, decode semantic information, and quantify uncertainty in the latent space.

LatentBKI is built on the intuition that neural network features are geometrically continuous and suitable for kernel methods. Interpolation is a common step in modern neural networks to infer features from geometrically adjacent points, used especially in up-sampling or deformable operations. Interpolation of point x on feature grid G with height H and width W can be written as:

$$G_x \approx \frac{\sum_{i=1}^{H} \sum_{j=1}^{W} w_{ij} G_{ij}}{\sum_i \sum_j w_{ij}} \tag{2.29}$$

where i and j are indices of neighboring cells, and weights w are determined by the distance of query point x to neighboring cells. This equation resembles the Nadaraya-Watson kernel estimate of the expected value, written similarly as:

$$\hat{G}_x = \frac{\sum_{i=1}^{N} k(x - x_i) G_i}{\sum_{i=1}^{N} k(x - x_i)} \tag{2.30}$$

for a set of N data points. As we will show next, our method produces an expectation equivalent to the Nadaraya-Watson kernel estimate, with the addition of quantifiable uncertainty through conjugate priors.

2.9.1 Latent Mapping Representation

Our map representation consists of a voxel map with voxels $*$ located at position x_*. At each time-step our map is provided a set of points $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^3$ is the position of point i and $y_i \in \mathbb{R}^C$ is the corresponding latent feature of point i. From these points, our goal is to probabilistically update the latent parameters of each voxel, y_*, to obtain the *posterior*.

In order to accomplish this goal, we first define a Gaussian likelihood over the feature space, such that:

$$p(y_i | \mu_i, \Sigma_i) = \mathcal{N}(y_i; \mu_i, \Sigma_i). \tag{2.31}$$

Since the features originate from a neural network, the likelihood defines an unknown expectation and variance which the point's features are sampled from. Similarly, we can define the points observed within a voxel $*$ according to the same likelihood distribution. From the likelihood, we can write an expression for the posterior of the latent parameters $\theta_* = \{\mu_*, \Sigma_*\}$ of voxel $*$ using Bayes' rule as:

$$p(\theta_* \mid x_*, \mathcal{D}) \propto p(\mathcal{D} \mid \theta_*, x_*) p(\theta_* \mid x_*). \tag{2.32}$$

In order to model the distribution over the parameters θ_* of the voxel, which themselves define a multivariate Gaussian distribution, we adopt the conjugate prior of the multivariate Gaussian distribution, the normal-inverse Wishart distribution. The normal-inverse Wishart distribution defines a distribution over the multivariate Gaussian with unknown mean μ_* and covariance Σ_* through parameters $\mu'_*, \Psi_*, \lambda_*, v_*$ as:

$$p(\mu_*, \Sigma_*) = \mathcal{N}\left(\mu_*; \mu'_*, \frac{\Sigma_*}{\lambda_*}\right) \cdot \mathcal{W}^{-1}(\Sigma; \Psi_*, v_*), \tag{2.33}$$

where the hyper-parameters represent the prior expectation of the mean (μ'_*), the expectation of the covariance (Ψ_*), and the confidence in the mean and covariance estimates (λ_*, v_*). In our case, λ_* and v_* are equal and correspond to weighted counts of the total observations.

Although the conjugate prior provides a closed-form solution for updating the multivariate normal distribution parameters for each voxel, it does not consider the spatial locations of points. Intuitively, points that are closer to the centroid of the voxel should have a higher influence, while points that are further from the voxel centroid should have a lower influence. Therefore, we adopt the solution of [191], which defines an extended likelihood distribution that considers the spatial relationship of points to voxels through the use of a kernel function $k(x_i, x_*)$ as:

$$p(y_i|x_i, \theta_*, x_*) \propto p(y_i|\theta_*)^{k(x_i, x_*)}. \tag{2.34}$$

The only requirements when defining the extended likelihood are that

$$k(x, x) = 1 \,\forall\, x \quad \text{and} \quad k(x, x_*) \in [0, 1] \,\forall\, (x, x_*). \tag{2.35}$$

Applied to the previously defined Gaussian likelihood, the extended likelihood can be written as:

$$p(y_i|\theta_*, x_i, x_*) = \mathcal{N}\left(y_i; \mu_*, \frac{\Sigma_*}{k(x_i, x_*)}\right). \tag{2.36}$$

Effectively, observations close to the voxel centroid have lower variance than observations of further points, resulting in more influence or gain. Following Semantic BKI, we use a symmetric sparse kernel [118] as the kernel function for direct comparison. Substituting the extended likelihood into (2.32), we can now define a spatial update over the voxel parameters as:

$$p(\theta_*|x_*, y_{1:N}, x_{1:N}) \propto \left[\prod_{i=1}^{N} p(y_i|\theta_*, x_i, x_*)\right] p(\theta_*|x_*). \tag{2.37}$$

Next, we present our map update algorithm, which follows the closed-form solution derived by [191].

2.9.2 Latent Mapping Update

First, we initialize the confidence over the mean and covariance of each voxel to a non-informative value of $\lambda_* \approx 0$. As points are observed, the value of λ_* increases, indicating more confidence in the expected mean and covariance of the semantics within the voxel. At time-step t, voxels are parameterized by prior μ_*^{t-1}, Ψ_*^{t-1} and confidence λ_*^{t-1}, with input points $\mathcal{D}$.

The influence of the new observations is calculated by:

$$\overline{k}_* = \sum_{i=1}^{N} k(x_*, x_i), \tag{2.38}$$

where the kernel function measures the influence of point i over voxel $*$. The confidence in the mean and covariance is updated:

$$\lambda_*^t = \lambda_*^{t-1} + \overline{k}_*. \tag{2.39}$$

Input observations are then used to compute the new mean as a running average:

$$\overline{y}_* = \sum_{i=1}^{N} \frac{k(x_*, x_i)}{\overline{k}_*} y_i, \quad \mu_*^t = \frac{\lambda_*^{t-1}\mu_*^{t-1} + \overline{y}_*\overline{k}_*}{\lambda_*^t}. \tag{2.40}$$

We note that the formulation of the new mean resembles the Nadaraya-Watson estimate mentioned in Section 2.9.1, as input features are weighted by the kernel function to obtain a weighted average.

Last, following [191], we update the expected covariance by weighting the covariance of the newly observed points:

$$\overline{E}_* = (\overline{y}_* - \mu_*^{t-1})(\overline{y}_* - \mu_*^{t-1})^T, \tag{2.41}$$

$$\overline{S}_* = \sum_{i=1}^{N} k(x_i, x_*)(y_i - \overline{y}_*)(y_i - \overline{y}_*)^T, \tag{2.42}$$

$$\Psi *^t = \Psi *^{t-1} + \overline{S}_* + \frac{\lambda_*^{t-1}\overline{k}_*}{\lambda_*^t}\overline{E}_*. \tag{2.43}$$

As new points are obtained, the above process is repeated to update the mean, covariance, and confidence level. For simplicity, we assume a diagonal covariance, which significantly reduces the number of parameters stored per voxel. Diagonal covariances are common in the feature space and are used in variational auto-encoders (VAEs).

2.9.3 Inference

After updating the map, we can compute an expectation and variance for features observed within each voxel. First, the distribution can be marginalized to obtain a posterior predictive solution that defines the probability of observing a feature y_* at voxel centroid x_*. The posterior predictive distribution for voxel $*$ is:

$$p(y_*|x_*) = t_{\lambda_*}\left(\mu_*, \frac{\lambda_* + 1}{\lambda_*^2}\Psi_*\right), \tag{2.44}$$

where t_{λ_*} is the multivariate Student-t distribution. The multivariate Student-t distribution has an expectation of $\mathbb{E}(y_*) = \mu_*$, when $\lambda_* > 1$, and a variance of $\text{Var}(y_*) = \frac{\lambda_*}{\lambda_* - 2}\Psi_*$, for $\lambda_* > 2$. However, both the expectation and variance are within the latent space of the neural network and must be decoded to obtain a meaningful interpretation.

When performing open-dictionary inference, the input is a set of phrases W defining semantic categories, which are processed by a language model to obtain text embeddings $F_w \in \mathbb{R}^C$ per phrase $w \in W$. Specifically, in our experiment, we encode each phrase w as a Clip feature vector F_w of 512 length $F_w = \text{Encoder}(w)$. Inspired by LSeg, we obtain the categorical prediction $\hat{w}$ as:

$$\hat{w} = \arg\max_w \frac{F_w^T \mu_*}{\|\mu_*\|_2 \|F_w\|_2}. \tag{2.45}$$

We note that while we present the decoding for open-dictionary queries, LatentBKI can be decoded into any format using neural network decoders.

2.9.4 Uncertainty Quantification

While the map update step described above can propagate uncertainty, the variance is limited to the latent space of the neural network encoder. Therefore, we propose two methods to quantify uncertainty.

First, following the approach of other neural network uncertainty quantification methods, we propose quantifying uncertainty through sampling. To quantify uncertainty through sampling, we sample many realizations of the voxel feature y_*, which we decode through the neural network decoder. Then, we compute the variance of the predictions in the decoded space. While this approach is accurate, it requires extra computation, which we propose to avoid.

Based on a common approach of information quantification in optimal experimental design [144], we can quantify uncertainty through p-optimality [97]. Although p-optimality leads to many solutions [143], in this work, we compute the uncertainty U_* via E-optimality criterion

as:

$$U_* = \max\left(\mathrm{Var}(y_*)\right), \tag{2.46}$$

and the uncertainty via D-optimality criterion as:

$$U_* = \exp\left(\frac{\sum_{i=1}^{C} \log\left(\mathrm{Var}(y_*)_i\right)}{C}\right), \tag{2.47}$$

where a large value corresponds to high uncertainty. This is because, experimentally, we find that E-optimality is highly correlated with the sampling-based uncertainty and is quick to compute. See Section 2.9.5.3 for detailed experiments. The p-, E-, and D-optimality criteria are discussed more in the literature [144] and describe means of computing heuristics to describe the uncertainty of vectors or matrices.

2.9.5 Results

We quantitatively and qualitatively show that LatentBKI effectively extends the continuous probabilistic mapping literature to neural network latent spaces, bringing quantifiable uncertainty and spatial smoothing to VL maps. First, we compare LatentBKI against closed-dictionary continuous mapping to verify that operations in the latent space of neural networks do not affect mapping performance. Next, we compare LatentBKI against VLMap, which performs latent space mapping but does not leverage spatial information or quantify uncertainty. Third, we study the correlation of our uncertainty quantification with segmentation errors and the effect of spatial smoothing. Finally, we conduct real-world experiments to demonstrate the ability of our map to transfer to real-world open-dictionary scenarios due to the strong generalization capabilities of large VL networks.

Table 2.6: Comparison against closed-dictionary BKI mapping.

Data	Method	Acc. (%)	mIoU (%)	Queries
Indoor	Segmentation	66.20	20.95	N/A
	ConvBKI (Single)	68.87	24.28	Fixed
	LatentBKI	68.16	23.14	**Open**
Outdoor	Segmentation	90.46	52.05	N/A
	ConvBKI (Single)	91.01	56.38	Fixed
	LatentBKI	91.29	57.33	**Open**

2.9.5.1 Comparison against categorical space mapping method

First, we compare LatentBKI against the closed-dictionary semantic mapping method defined in [62], which leverages BKI with a categorical likelihood to update the map. Specifically, we compare against ConvBKI [199] using a single un-trained spherical kernel for direct comparison. Our goal of this study is to verify that LatentBKI can generalize BKI into the latent space of neural networks without any significant changes in performance.

We evaluate LatentBKI and ConvBKI (single) on an indoor and outdoor dataset. For the indoor dataset, we use Matterport3D (MP3D) [26] with an image semantic segmentation network, Language-driven Semantic Segmentation (LSeg) [109]. For the outdoor dataset, we choose the Semantic KITTI dataset [14] and use the Sparse Point-Voxel Convolution Neural Network (SPVCNN) [178, 206] for point cloud semantic prediction. Since the latent space of LSeg has a large dimension of 512, we use Principal Component Analysis (PCA) to down-sample features for the map update. The PCA algorithm is trained on held-out MP3D sequences we do not compare with ConvBKI on, and we experimentally find that reducing the size of the latent space of LSeg to 64 results in only a 0.09% loss in accuracy and 0.42% decrease in mIoU.

We apply the same configuration to both mapping algorithms to ensure comparable results. Each algorithm uses a voxel resolution of 0.1 meters, a sparse kernel with a kernel length of 0.5 meters, and a filter size of 3, determining how many neighboring voxels should be updated for a single-point observation. We apply a cubic filter in this case, so $3^3 = 27$ neighboring voxels will be considered. Since both methods compare spatial smoothing, we provide 80% of the points as input and evaluate semantic predictions over the mean intersection over union (mIoU) and accuracy metrics on the remaining 20% of the points.

As shown in Table 2.6, LatentBKI performs similarly to ConvBKI over indoor and outdoor datasets. These results verify that BKI with a Gaussian likelihood is suitable for the latent space of multiple neural networks and does not cause any significant changes in performance. While LatentBKI results in a marginal improvement in quantitative performance on the outdoor data, the slight decrease in indoor data is due to the dimensionality reduction applied by PCA on the input to LatentBKI. Additionally, although the studies are performed on closed-dictionary segmentation, LatentBKI enables open-dictionary continuous mapping as the per-voxel latent features can be decoded into more meaningful outputs. Next, we compare LatentBKI with a popular latent mapping algorithm, VLMap.

Table 2.7: Comparison against VLMap, which performs latent mapping without spatial smoothing or uncertainty quantification. Results are compared on scenes gTV8FGcVJC9 (1) and 5LpN3gDmAk7 (2) of MP3D.

Scene	Method	Acc. (%)	mIoU (%)	Uncertainty
1	Segmentation	61.08	19.57	N/A
	VLMap	60.94	20.32	No
	LatentBKI	**63.30**	**21.56**	**Yes**
2	Segmentation	63.27	13.12	N/A
	VLMap	64.12	12.44	No
	LatentBKI	**66.10**	**14.92**	**Yes**

2.9.5.2 Latent Mapping Comparison

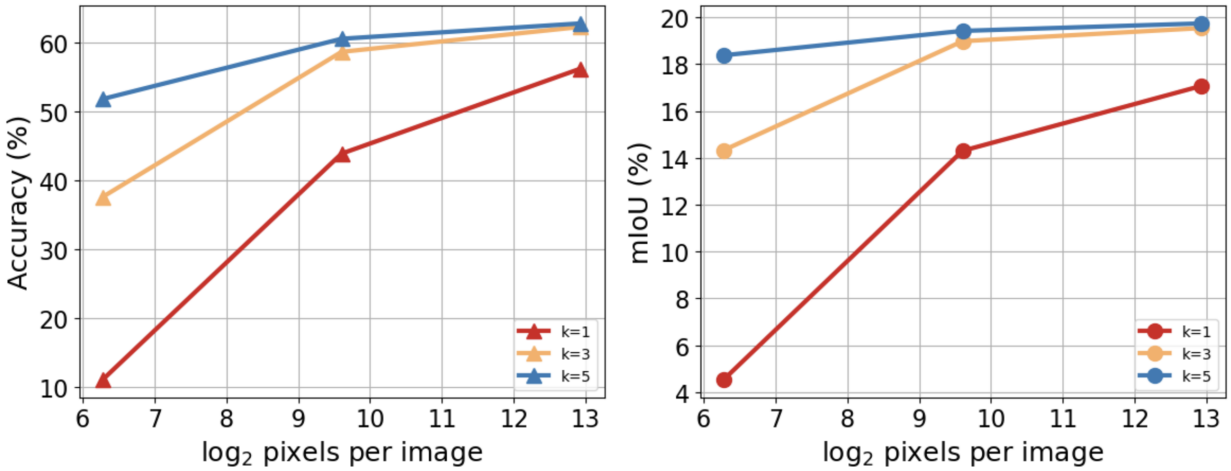
We compare LatentBKI with a similar open-dictionary mapping method VLMap [85], which also updates voxels through a weighted averaging approach. Since VLMap does not utilize spatial information, we compare LatentBKI to VLMap with a filter size of 1, disregarding points in nearby voxels when calculating the posterior state of a voxel. Both methods use the open-dictionary encoder LSeg, with features downsampled by PCA to a latent size of 64.

We compare each method on the MP3D dataset [26] following the same experimental setup as VLMap, including a resolution of 0.05 m to account for the fine-resolution indoor environment. To ensure that the input and evaluation points are consistent, we implemented the pre-processing heuristics of VLMap within our method. These heuristics include discarding pixels with extreme depths, discarding points outside of the scene range, and downsampling input points. Specifically, the same set of 1% of remaining pixels are provided to each method, and points with depth less than 0.1 meters or greater than 6 meters are excluded. By following the same downsampling heuristics as VLMap in our evaluation, we isolate the effect of the sparse kernel function used by our method to weight input points compared to the depth-wise weighting scheme employed by VLMap.

The results of our experiments on two different MP3D sequences, as shown in Table 2.7, indicate that LatentBKI outperforms VLMap in both accuracy and mean IoU, which we attribute to the sparse kernel. At its core, the VLMap algorithm is similar to our method of calculating the expectation with a different weighting scheme, and without spatial smoothing or quantifiable uncertainty. This performance improvement may be attributed to the formalized mathematical updates in LatentBKI, which provide a more robust integration of observations and better handling of uncertainty. Additionally, the formal probabilistic approach we leverage in LatentBKI allows for quantifiable uncertainty, which we study next.

2.9.5.3 Ablation Studies

Two benefits of LatentBKI are the spatial smoothing effect of continuous mapping and the ability to quantify the temporal uncertainty of neural network predictions. In this section, we study the quantitative improvement from different kernel sizes, as well as compare the sampling and P-Optimality methods for quantifying uncertainty.



(a) Accuracy vs. Input Sparsity

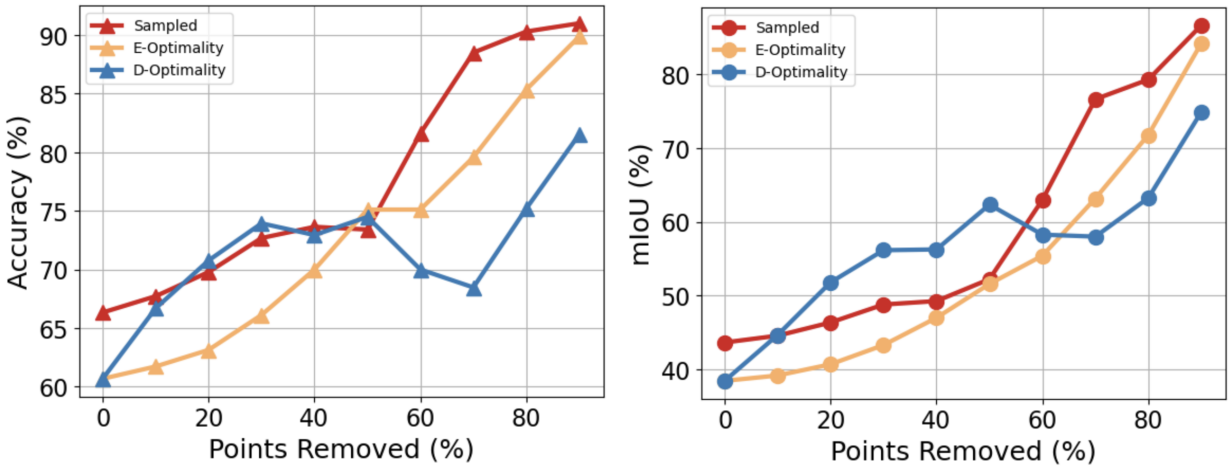
(b) mIou vs. Input Sparsity

Figure 2.10: Effect of spatial smoothing with varying levels of image sparsity. Spatial smoothing, indicated by the filter size k, is most effective for sparse images. The original image has a resolution of 720 by 1080 pixels.

Spatial Smoothing: In real-world applications, data is often sparse due to sensors such as LiDAR or sparse stereo matching algorithms. BKI provides a probabilistic technique to create more complete maps from sparse spatial data by leveraging the spatial smoothing effect of kernels. In this experiment, we compare different kernel sizes and their ability to complete the map from sparse data.

All kernels are compared on the same scene of MP3D, where data is downsampled temporally to incorporate only one in 3 frames and at the image level to use a randomly sampled set of pixels from each image. Fig. 2.10 demonstrates plots of the segmentation performance of different kernel sizes and varying sparsity levels. At extreme sparsity levels, spatial smoothing of points benefits the map completeness. As the input becomes more dense, in this case, due to high-resolution images with accurate depth, the effect of the kernel is diminished.

Uncertainty Quantification: To evaluate the ability of LatentBKI to quantify uncertainty meaningfully, we quantitatively and qualitatively compare uncertainty quantification on the MP3D dataset using LSeg as the encoder. We construct a map using LatentBKI, then compare uncertainty quantified using D-optimality, E-optimality, and sampling as described



(a) Accuracy vs. uncertainty

(b) mIou vs. uncertainty

Figure 2.11: Sparsification plot of segmentation performance compared to quantified uncertainty. As uncertain points are removed, a well calibrated uncertainty should cause the segmentation performance to increase.

in Section 2.9.4. Whereas the sampling-based method is commonly used, it is computationally expensive compared to the E-optimality and D-optimality-based techniques.

To quantitatively compare each uncertainty quantification method, we create sparsification plots [86] identifying the correlation between uncertainty and prediction error, shown in Fig. 2.11. To create the sparsification plots, we sort points in a test set by the predicted uncertainty. Next, we separate the sorted points into bins and iteratively remove the most uncertain bin. If the uncertainty is properly calibrated, we expect to see an increase in the accuracy as uncertain points are removed. As seen in Fig. 2.11, both the accuracy and mIoU metrics are correlated with all three methods, especially sampling and E-optimality. In addition to a strong correlation between latent uncertainty and error in the decoded predictions, E-optimality benefits from efficient computation.

We also qualitatively compare uncertainty quantification between sampling and E-optimality, shown in Fig. 2.12. We observe that the most uncertain voxels are typically located at the edges of rooms or at objects that are difficult for the VL network to identify due to ambiguity or poorly captured images. Similar to the sparsification plots, we find that E-optimality closely aligns with the uncertainty estimated through sampling, indicating that E-optimality is an effective approximation for the latent uncertainty.

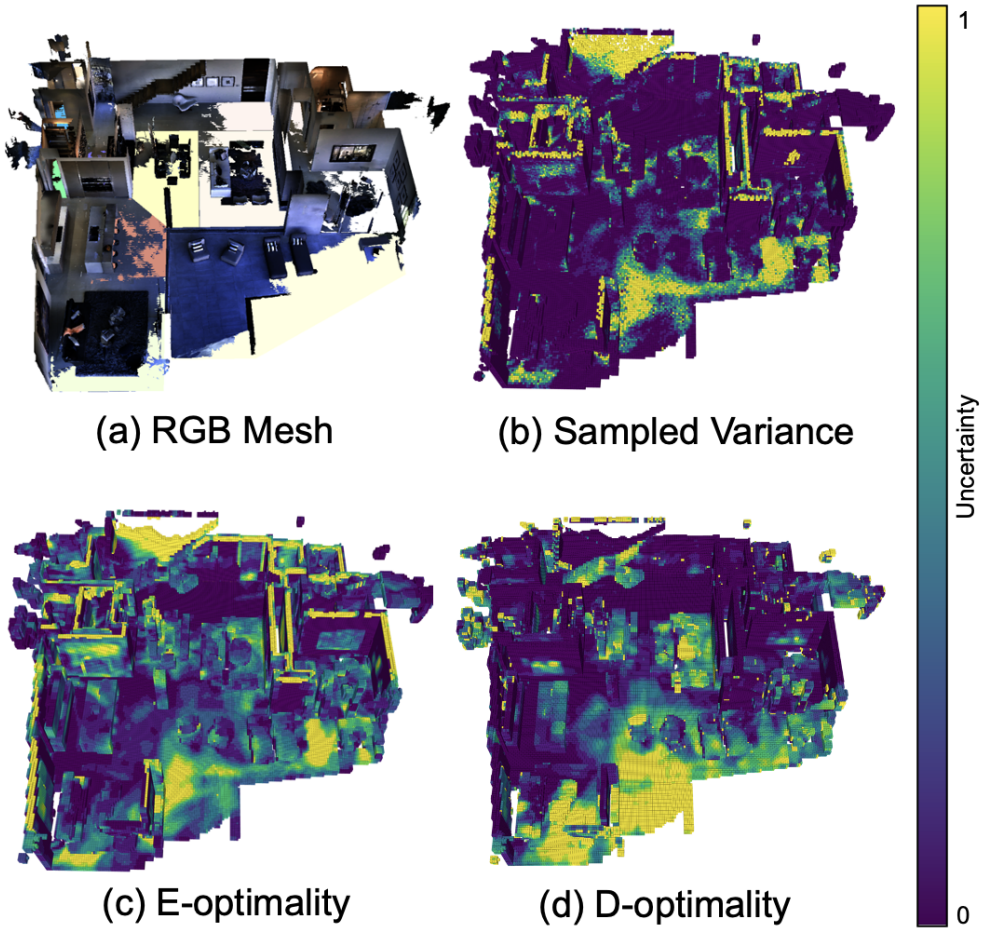


Figure 2.12: Uncertainty maps for 5LpN3gDmAk7 MP3D sequence. (a) Covered house mesh in the sequence. (b) Categorical variance map by sampling from distribution. (c) Variance map by using E-optimality in latent space. (d) Variance map by using D-optimality in latent space.

2.10 Conclusions

We propose a Bayesian inference framework for real-time elevation and voxel mapping and terrain property estimation from RGB-D images. Our method outperforms other algorithms both in simulation and the real-world. Unlike traversability methods, our algorithm provides terrain property information that can enable robots to adjust their locomotion to traverse regions of low friction rather than just avoid them.

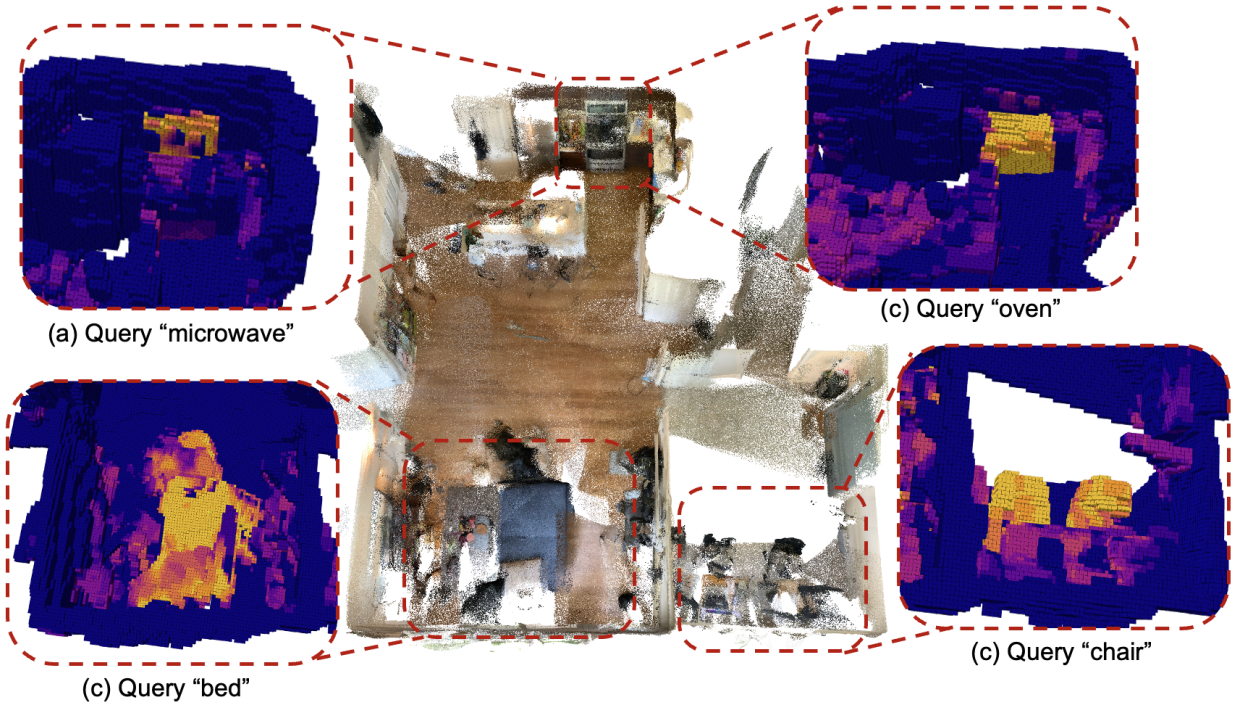


Figure 2.13: Open vocabulary query task results. Query results are shown in heat map, brighter colors indicates higher values.

CHAPTER 3

Multi-Modal Bayesian Inference for Semantic Classification and Property Prediction

Scene understanding from exteroceptive sensing via images or point clouds enables mobile robots to perform object avoidance, terrain traversal, and a variety of other tasks. The introduction of high-level and task-dependent semantic labels for scene understanding has recently spurred rapid growth in this area [76, 189]. To extract semantic labels from images or point clouds, it is common to employ semantic segmentation neural networks; however, even state-of-the-art methods return inconsistent labels under viewpoint or lighting changes, and are heavily reliant on the data used for training. Unfortunately, collecting data to train these networks is expensive and out-of-distribution examples may lead to incorrect classifications.

To address the challenge of viewpoint inconsistency, several state-of-the-art methods project image-based semantic classifications onto geometric maps [61, 115]. This provides a shared frame of reference between images thereby enabling the application of probabilistic methods to fuse semantic data. While such methods demonstrate improved performance over state-of-the-art image-based segmentation [47], these methods have been restricted to either visual or lidar data. This is limiting since the properties of a semantic class are often of primary interest rather than the class labels themselves while completing a robotic task. For instance, the coefficient of friction is more informative than the terrain class for a legged robot. While recent semantic mapping methods have derived recursive filters for image-based semantic predictions, properties conditioned on these semantic classes are often assumed to be known *a priori* and are considered immutable at run time [23, 47, 61].

This chapter introduces a multi-modal semantic prediction and property estimation framework to jointly estimate semantic labels alongside physical properties as depicted in Figure 3.1. The key insight is that semantic classifications may be conditioned on their physical properties and physical properties may likewise be conditioned on their semantic classifications. In this way, vision-based semantic classifications can be used to build a measurement



Figure 3.1: The method proposed in this chapter jointly estimates semantic classifications and physical properties by combining visual and tactile data into a single semantic mapping framework. RGB-D images are used to build a metric-semantic map that iteratively estimates semantic labels. A property measurement is taken which in turn updates both the semantic class predictions and physical property estimates. In the depicted example, the robot is unsure if the terrain in front of it is snow or ice from vision measurements alone (prior estimates) which dramatically affects the coefficient of friction and the associated gait that can be applied to safely traverse the terrain. The robot uses a tactile sensor attached to its manipulator to update its coefficient of friction estimation (posterior estimates), which then enables it to change gaits to cross the ice safely.

model for physical properties, and tactile-based sensing can be used to update the semantic predictions via property measurements. To apply this insight, this chapter demonstrates how to recursively estimate physical properties from semantic class predictions using only visual information. Subsequently, this chapter describes how to exploit a tactile sensing modality to update the semantic classification and property estimates probabilistically and in closed-form using Bayesian inference.

The efficacy of the proposed approach is demonstrated on several hardware platforms including a robotic manipulator and a legged robot. First, we show that by leveraging friction measurements obtained through tactile sensing we can improve semantic classification performance when compared to state-of-the-art semantic mapping methods that rely solely on visual information. Second, we demonstrate that our approach estimates physical properties more accurately than a method using vision alone. Multiple sensing modalities are used to estimate the coefficient of friction of the terrain surrounding a robot, which enables the use of an online risk-aware planner that switches the legged robot from a dynamic gait to a static, stable gait if the expected value of the coefficient of friction of the terrain falls below a given threshold. Importantly, the proposed multi-modal framework allows the legged robotic system to complete a challenging terrain traversal task whereas state-of-the-art vision-based traversability estimation methods fail to accurately predict the traversability of the terrain resulting in the robot slipping and falling. Finally, we show the broad applicability of the proposed framework for affordance-based property prediction by demonstrating door opening with unknown push-pull interactions.

The contribution of this work is the introduction of a probabilistic model for jointly estimating semantic classifications and physical properties. This enables closed-form Bayesian updates given visual semantic classifications and tactile property measurements. We demonstrate that by leveraging this probabilistic approach, we are able to improve semantic prediction accuracy by an average of 21.3% in simulation using only one property measurement. These results are validated in hardware demonstrations. Two case studies are subsequently presented that illustrate the utility of performing this type of multi-modal property estimation. An open-source C++ and ROS package for the robot-agnostic implementation of the proposed approach as well as videos of the case studies are provided on the project page website[1].

The remainder of this chapter is organized as follows: Section 3.1 summarizes the semantic prediction and property estimation literature and Section 3.2 reviews preliminary material used throughout the chapter. Section 3.3 provides an overview of our method. Section 3.4 introduces the vision-based semantic prediction pipeline and Section 3.5 the tactile-based semantic prediction and property estimation pipeline. Section 3.6 describes the implementation. Sections 3.7 and 3.8 describe the evaluation of our algorithm in simulation and using hardware demonstrations, respectively. Section 3.9 discusses future work and provides concluding remarks.

[1] https://roahmlab.github.io/multimodal_mapping/

3.1 Related Works

This section describes related work in semantic prediction and property estimation as well as the gap in the literature this work addresses.

3.1.1 Semantic Prediction

State-of-the-art methods for semantic classification almost exclusively rely on neural networks to produce pixel-wise semantic predictions [76, 77, 171]. Often these approaches use task-dependent object designations as semantic labels [38, 110] such as abstract topological information [105] or material classifications [186]. Unfortunately, semantic prediction methods often have no temporal consistency, meaning two subsequent images in a video stream may have vastly different semantic predictions.

To overcome this challenge and relate semantic predictions spatially and temporally, early work in semantic mapping projected semantic predictions onto a geometric representation forming a metric-semantic map [103]. In this way, image-based semantic predictions have a shared frame of reference within this map. These early methods employed voting-like schemes, regularization, and other algorithmic methods to fuse semantic predictions [115, 175].

Recent work in semantic mapping treats the semantic fusion process probabilistically to recursively estimate semantic labels [61, 198]. These methods treat pixel-wise semantic predictions as measurements and probabilistically fuse these measurements over the map using Bayesian inference. By approaching semantic prediction in this way, these approaches are robust to noisy pixel-wise semantic classifications and have been shown to be more accurate than methods that rely on image-based classifications alone [47].

Multi-modal semantic prediction, and the corresponding semantic mapping extension, have been around since the rise of modern computer vision [37]. Much of this work comes from the vision community and, as such, many of these methods rely on different vision-based modalities such as depth [193], thermal imaging [173], and scale-invariant feature transform (SIFT) factors [166]. Many modern multi-modal semantic classification methods are learning-based, where the multi-modal visual features are combined either at the input of, or downstream within, neural networks to output semantic classifications [98, 188].

3.1.2 Property Estimation

While semantic classification methods have seen widespread use in the robotics community, there are many robotic tasks (e.g., footstep planning, grasp planning, manipulation) where

the underlying physical properties of the semantic classes are critical. Additionally, if semantic classifications are incorrect, the resultant property estimates may also be incorrect, leading to task failure.

Previous work in property estimation has focused on estimating the expected value for the coefficient of friction from vision alone [18, 133, 195]. These methods have been extended in a recent paper that estimates probability distributions for the coefficient of friction given visual semantic class estimates by conditioning properties on semantic classes [47].

Other methods for property estimation focus specifically on estimating traversability [61, 139]. This is done to bypass the need to estimate terrain properties by instead estimating the regions in the environment a robot can traverse using visual features [54]. This is often an ill-posed problem as traversability is not only terrain-dependent, but also dependent on the motion of the robot (e.g., walking vs. running) [100]. For example, it is possible for small insects to walk across a large body of water, but if a large robot applied the same gait it would most likely sink.

Notably, there has been a recent shift towards multi-modal property estimation in the literature. One such approach proposes an online, self-supervised learning framework to estimate traversability by back-projecting previously traversed regions into the camera frame and training a network in a self-supervised manner to correctly label these previously seen regions as traversable [55]. This method is considered multi-modal because it incorporates an implicit goal-success signal, in the form of the robot having not fallen over, alongside visual information. Other work aims to encode property features in a latent space using tactile and visual information to train reinforcement learning agents [163]. Recently, [112] proposed a method for learning motion policies by training a network to relate image-based features and physical properties such as friction in simulation. It was shown that the trained network accurately predicted properties in hardware demonstrations, however these predictions only output a single scalar value for properties without considering uncertainty in the prediction. A similar framework for multi-modal probabilistic property estimation within a semantic mapping framework was also demonstrated for the purposes of traversability estimation [23]. In this case, the terrain classes are modelled probabilistically using Dirichlet distributions (analogous to [47, 61]) and trained networks are then used to predict the likelihood of slipping using data collected offline.

3.1.3 Gap in the Literature

The methods discussed in the prior two subsections have two notable shortcomings. The first is that, while existing property estimation algorithms condition physical properties on

semantic classifications, this conditioning of properties is immutable, meaning the property values of semantic classes cannot be updated during operation [23, 47, 61]. This is undesirable especially when such physical properties have intra-class variation or there are errors in these initial estimates. For example, the coefficient of friction of ice depends on the conditions of ice formation, outdoor temperature, and the presence of snow on the surface of the ice. Assuming that the coefficient of all ice is identical may result in unsafe or overly conservative behavior.

The second short-coming of these methods is the one-way conditioning of physical properties on semantic classes. This can be unduly limiting because visual information alone may not be sufficiently informative. For example, in Fig. 3.1 it is challenging to differentiate between snow and ice from just an RGB sensor. Using a tactile sensor in this instance to estimate the friction coefficient can inform and improve the accuracy of the semantic classification. Using physical properties to inform semantic classification can enable the application of multiple sensing modalities while providing a means of verification for semantic predictions output by trained networks. This improves the accuracy of semantic predictions without requiring new training data and network retraining.

To address these short-comings, this chapter proposes a method to *jointly* estimate semantic classifications and physical properties online using Bayesian inference. Semantic classifications from vision enable the prediction of physical properties, and tactile measurements enable the online update of physical property estimates. This in turn enables the semantic classification likelihoods to be updated.

3.2 Preliminaries

This section introduces notation, moments, and conjugate pairs. Vectors, written as columns, are typeset in bold and lowercase, while sets and matrices are typeset in uppercase. The element i of a vector $\boldsymbol{x}$ is denoted as x_i. An n-dimensional closed interval is denoted by $[a, b]^n$. Let p denote a normalized probability mass or density function. Following the convention of [182], we distinguish between different probability mass or density functions by the variable used as an input argument into each p.

3.2.1 Conjugate Pairs

In probability, conjugate distributions are a pair of distributions such that, given a prior and a likelihood, the posterior computed using Bayes' theorem belongs to the same family of distributions as the prior [39]. This enables tractable computation of closed-form solu-

tions for the posterior [4]. We introduce two such conjugate pairs here: the Dirichlet and Categorical distributions, and the Dirichlet Normal-Inverse-Gamma product and Gaussian mixture distributions.

3.2.1.1 First Conjugate Pair

Let $z \in \{1, \ldots, k\}$ denote a discrete random variable. The probability mass function of the Categorical distribution represents the probability that a sample z belongs to class i:

$$p(z = i|\boldsymbol{\theta}) = Cat(z = i|\boldsymbol{\theta}) = \theta_i, \tag{3.1}$$

where $i \in \{1, 2, \ldots, k\}$, $\boldsymbol{\theta} \in [0, 1]^k$, and $\sum_i \theta_i = 1$.

The Dirichlet distribution is a continuous k-variate probability distribution that is parameterized by a vector $\boldsymbol{\alpha} \in \mathbb{R}^k_{\geq 0}$ with probability density:

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = Dir(\boldsymbol{\theta}|\boldsymbol{\alpha}) \tag{3.2}$$

$$= \frac{\Gamma(\sum_{j=1}^k \alpha_j)}{\sum_{j=1}^k \Gamma(\alpha_j)} \prod_{j=1}^k \theta_j^{\alpha_j - 1}, \tag{3.3}$$

where

$$\Gamma(\alpha_j) = \int_0^\infty x^{\alpha_j - 1} \exp(-x) dx. \tag{3.4}$$

The Dirichlet distribution is a conjugate prior to the Categorical distribution as is formalized in the following theorem:

Theorem 3. *Let $\mathcal{Z} = \{z_1, \ldots, z_n\}$ be a set of measurements drawn from a Categorical distribution, $p(z_j = i|\boldsymbol{\theta})$, and let the prior for $\boldsymbol{\theta}$ be a Dirichlet distribution, $p(\boldsymbol{\theta}|\boldsymbol{\alpha})$. The posterior computed using Bayes' theorem, $p(\boldsymbol{\theta}|\mathcal{Z}, \boldsymbol{\alpha})$, is also a Dirichlet distribution such that:*

$$p(\boldsymbol{\theta}|\mathcal{Z}, \boldsymbol{\alpha}) = Dir(\boldsymbol{\theta}|\tilde{\boldsymbol{\alpha}}), \tag{3.5}$$

where

$$\tilde{\alpha}_j = \alpha_j + \sum_{z_i \in \mathcal{Z}} 1\{z_i = j\}, \tag{3.6}$$

and $1\{z_i = j\}$ is equal to 1 when the expected class of measurement z_i is class j and is zero otherwise.

Proof. Let $p(\boldsymbol{\theta}|\boldsymbol{\alpha})$ be the prior belonging to the family of Dirichlet distributions and $p(z|\boldsymbol{\theta})$

the likelihood belonging to the family of Categorical distributions. Let $\mathcal{Z} = \{z_1, \ldots, z_n\}$ be a set of measurements drawn from the Categorical likelihood.

The Law of Total Probability is used to compute the posterior distribution:

$$p(\boldsymbol{\theta}|\mathcal{Z}, \boldsymbol{\alpha}) \propto p(\boldsymbol{\theta}, \mathcal{Z}, \boldsymbol{\alpha}) \tag{3.7}$$

$$= p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \prod_{z_i \in \mathcal{Z}} p(z_i|\boldsymbol{\theta}), \tag{3.8}$$

where the equality is due to conditional independence.

Then, substituting (2.2) and (2.3) into (3.8), we arrive at:

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \prod_{z_i \in \mathcal{Z}} p(z_i|\boldsymbol{\theta}) \propto \prod_{j=1}^{k} \theta_j^{\alpha_j - 1} \prod_{z_i \in \mathcal{Z}} \prod_{j=1}^{k} \theta_j^{1\{z_i = j\}} \tag{3.9}$$

$$= \prod_{j=1}^{k} \theta_j^{\alpha_j - 1 + \sum_{z_i \in \mathcal{Z}} 1\{z_i = j\}}. \tag{3.10}$$

This is exactly the Dirichlet distribution with parameters

$$\tilde{\alpha}_j = \alpha_j + \sum_{z_i \in \mathcal{Z}} 1\{z_i = j\}. \tag{3.11}$$

□

Theorem 3 enables the computation of the predictive posterior. In particular, one of the goals of this chapter is to predict the probability that a new measurement belongs to class i given prior measurements $\mathcal{Z}$:

$$p(z = i|\mathcal{Z}, \boldsymbol{\alpha}) = p(z = i|\tilde{\boldsymbol{\alpha}}) \tag{3.12}$$

$$= \int_{\boldsymbol{\theta}} p(z = i|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\tilde{\boldsymbol{\alpha}}) \mathrm{d}\theta, \tag{3.13}$$

where $p(z = i|\boldsymbol{\theta})$ represents the Categorical likelihood and $p(\boldsymbol{\theta}|\tilde{\boldsymbol{\alpha}})$ the Dirichlet posterior probability. Computing this integral exactly is challenging. Fortunately, the theory of conjugate priors enables a closed-form solution [184, (3)]:

$$p(z = i|\tilde{\boldsymbol{\alpha}}) = \frac{\tilde{\alpha}_i}{\sum_{j=1}^{k} \tilde{\alpha}_j}. \tag{3.14}$$

3.2.1.2 Second Conjugate Pair

Let $\psi \in \mathbb{R}$ denote a continuous random variable. A Gaussian mixture is a continuous probability density function

$$p(\psi|\Theta) = \sum_{i=1}^{k} w_i \mathcal{N}(\psi|\mu_i, \sigma_i^2), \tag{3.15}$$

where $\Theta = \{w_i, \mu_i, \sigma_i^2\}_{i=1}^k$ and $\sum_{i=1}^k w_i = 1$. Next, we present a candidate for the conjugate prior of the Gaussian mixture model by conditioning the parameters, Θ, on a set of hyperparameters, Ψ. Previously, the Dirichlet distribution was used to parameterize these variables to form a conjugate pair. We perform a similar trick here using the Dirichlet Normal-Inverse-Gamma product distribution as the prior on Θ.

The Dirichlet Normal-Inverse-Gamma product distribution is defined as

$$p(\Theta|\Psi) = Dir(\boldsymbol{w}|\boldsymbol{a}) \prod_{i=1}^{k} \mathcal{N}\Gamma^{-1}(\mu_i, \sigma_i^2|\tau_i, \kappa_i, \beta_i, \gamma_i), \tag{3.16}$$

where $\Psi = \{a_i, \tau_i, \kappa_i, \beta_i, \gamma_i\}_{i=1}^k$, $\kappa_i, \beta_i, \gamma_i > 0$, and the Normal-Inverse-Gamma distribution is defined as

$$\mathcal{N}\Gamma^{-1}(\mu_i, \sigma_i^2|\tau_i, \kappa_i, \beta_i, \gamma_i) = \frac{\sqrt{\kappa_i}}{\sqrt{2\pi\sigma_i^2}} \frac{\gamma_i^{\beta_i}}{\Gamma(\beta_i)} \frac{1}{\sigma_i^2}^{\beta_i+1} \exp(-\frac{2\gamma_i + \kappa_i(x-\tau_i)^2}{2\sigma_i^2}). \tag{3.17}$$

When a measurement ψ is taken, we assume it is drawn from the Gaussian mixture measurement likelihood. The posterior of Θ is then computed using the following theorem from [90, (6)]:

Theorem 4. *Let ψ be a measurement drawn from a Gaussian mixture $p(\psi|\Theta)$. Let the prior for Θ be a Dirichlet Normal-Inverse-Gamma product distribution. Then the posterior computed using Bayes' theorem, $p(\boldsymbol{\theta}|\psi, \Psi)$, is:*

$$p(\Theta|\psi, \Psi) = \tfrac{1}{M} \textstyle\sum_{j=1}^{k} c_j Dir(\boldsymbol{w}|\tilde{\boldsymbol{a}}_j) \cdot \mathcal{N}\Gamma^{-1}(\mu_j, \sigma_j^2|\tilde{\tau}_j, \tilde{\kappa}_j, \tilde{\beta}_j, \tilde{\gamma}_j) \cdot \prod_{i \neq j}^{k} \mathcal{N}\Gamma^{-1}(\mu_i, \sigma_i^2|\tau_i, \kappa_i, \beta_i, \gamma_i), \tag{3.18}$$

where

$$\tilde{a_j} = a_j + 1, \tag{3.19}$$

$$\tilde{\tau_j} = \frac{\kappa_j \tau_j + \psi}{\kappa_j + 1}, \tag{3.20}$$

$$\tilde{\kappa_j} = \kappa_j + 1, \tag{3.21}$$

$$\tilde{\beta_j} = \beta_j + \frac{1}{2}, \tag{3.22}$$

$$\tilde{\gamma_j} = \gamma_j + \kappa_j \frac{(\psi - \tau_j)^2}{2(1 + \kappa_j)}, \tag{3.23}$$

$$c_j = \sqrt{\frac{\kappa_j}{\tilde{\kappa_j}}} \frac{\Gamma(\tilde{\beta_j})}{\Gamma(\beta_j)} \frac{{\gamma_j}^{(\beta_j)}}{\tilde{\gamma_j}^{(\tilde{\beta_j})}}, \tag{3.24}$$

$\tilde{\boldsymbol{a}}_j = \{a_1, \ldots a_{j-1}, \tilde{a}_j, a_{j+1}, \ldots, a_k\}$ *and* M *is the normalizing factor.*

Using Theorem 4, note the posterior, (3.18), is not in the same family of distributions as the prior, (3.16). More troublingly, the number of terms in (3.18) grows exponentially with the number of measurements. Rather than apply Theorem 4, we implement the method of moments described in the next section to approximate (3.18) as a Dirichlet Normal-Inverse-Gamma product distribution.

3.2.2 Method of Moments

Let $\boldsymbol{\lambda} = \{\lambda_1, \ldots, \lambda_n\}$ be an n-dimensional continuous random variable with probability density function $p(\boldsymbol{\lambda}|\xi)$ conditioned on parameters ξ. The i-th order moment is defined as

$$M_{g_i(\boldsymbol{\lambda})}(p) = \mathbb{E}[g_i(\boldsymbol{\lambda})] \tag{3.25}$$

where g_i is a monomial of $\boldsymbol{\lambda}$ of degree i. For some distributions, there exists a finite set of such moments, termed the *sufficient moments* [90] that fully define the set of parameters ξ. This means that one can construct the probability density function for this distribution by just using the sufficient moments. We denote the set of sufficient moments of a distribution p as $\mathbb{S}_p$.

The method of moments is a technique that fits a distribution to a set of sampled data by matching the sufficient moments of the distribution with the empirical moments computed from data. This approach may also be used to approximate a distribution by another distribution in a different family of distributions by matching sufficient moments of the first distribution with those of the second distribution. For our purposes, we use the method of

Algorithm 4: Method of Moments

Requires : Dataset $\mathcal{D} = \{\psi_l\}_{l=1}^N$, Prior $p(\Theta|\Psi)$

for $\psi \in \mathcal{D}$ **do**

- Compute $p(\Theta|\psi, \Psi)$ // (3.18)
- $\mathbb{E}[g_j(\boldsymbol{\theta})] \leftarrow \text{moments}(p(\Theta|\psi, \Psi))$ // (3.25)
- $\hat{\Psi} \leftarrow \text{matchMoments}()$ // (3.27)-(3.31)
- Compute $p(\Theta|\hat{\Psi})$

moments to approximate the posterior computed using Theorem 4 with a probability density from the family of Dirichlet Normal-Inverse-Gamma product distributions. This may be thought of as projecting the first distribution onto the family of the second distribution [90]. This process is summarized in Algorithm 4 for projecting (3.18) onto the family of distributions of (3.16).

3.2.3 Revisiting the Dirichlet Normal-Inverse-Gamma Product

By implementing Algorithm 4, we project (3.18) onto the family of Dirichlet Normal-Inverse-Gamma product distributions resulting in an approximate posterior. To compute this projection using the method of moments, we first need the sufficient moments of the Dirichlet Normal-Inverse-Gamma product distribution. From [90, §4.1], these sufficient moments are $\mathbb{S}_p = \{\mu_i, \sigma_i^2, \sigma_i^4, \mu_i^2\sigma_i^2, w_i, w_i^2\}_{i=1}^k$. This process is summarized in the following theorem:

Theorem 5. *Let ψ be a measurement drawn from a Gaussian mixture $p(\psi|\Theta)$. Let the prior for Θ be a Dirichlet Normal-Inverse-Gamma product distribution. Then the posterior computed using Bayes' theorem, $p(\boldsymbol{\theta}|\psi, \Psi)$, and projected onto the family of Dirichlet Normal-Inverse-Gamma product distributions via Algorithm 4 yields:*

$$p(\Theta|\hat{\Psi}) = Dir(\boldsymbol{w}|\hat{\boldsymbol{a}}) \prod_{i=1}^{k} \mathcal{N}\Gamma^{-1}(\mu_i, \sigma_i^2|\hat{\tau}_i, \hat{\kappa}_i, \hat{\beta}_i, \hat{\gamma}_i). \tag{3.26}$$

The parameters $\hat{\Psi}$ are computed using the sufficient moments of (3.18)*:*

$$\hat{\tau}_i = \mathbb{E}[\mu_i], \tag{3.27}$$

$$\hat{\kappa_i} = \frac{1}{\mathbb{E}[\mu_i^2\sigma_i^2] - \mathbb{E}[\mu_i]^2\mathbb{E}[\sigma_i^2]}, \tag{3.28}$$

$$\hat{\beta}_i = \frac{\mathbb{E}[\sigma_i^2]^2}{\mathbb{E}[\sigma_i^4] - \mathbb{E}[\sigma_i^2]^2}, \tag{3.29}$$

$$\hat{\gamma}_i = \frac{\mathbb{E}[\sigma_i^2]}{\mathbb{E}[\sigma_i^4] - \mathbb{E}[\sigma_i^2]^2}, \tag{3.30}$$

$$\hat{a}_i = \mathbb{E}[w_i]\frac{\mathbb{E}[w_i] - \mathbb{E}[w_i^2]}{\mathbb{E}[w_i^2] - \mathbb{E}[w_i]^2}. \tag{3.31}$$

Proof. Let $p(\Theta|\Psi)$ be the prior belonging to the family of Dirichlet Normal-Inverse-Gamma distributions and $p(\psi|\Theta)$ the likelihood belonging to the family of Gaussian mixtures.

From Theorem 4, the posterior of Θ is given by:

$$p(\Theta|\psi,\Psi) = \tfrac{1}{M}\textstyle\sum_{j=1}^{k} c_j Dir(\boldsymbol{w}|\tilde{\boldsymbol{a}}_j)\cdot\mathcal{N}\Gamma^{-1}(\mu_j,\sigma_j^2|\tilde{\tau}_j,\tilde{\kappa}_j,\tilde{\beta}_j,\tilde{\gamma}_j)\cdot\textstyle\prod_{i\neq j}^{k}\mathcal{N}\Gamma^{-1}(\mu_i,\sigma_i^2|\tau_i,\kappa_i,\beta_i,\gamma_i), \tag{3.32}$$

with parameters computed as per (3.19)-(3.24).

It is evident that the number of terms in this posterior grows exponentially for each measurement. To overcome this challenge, we project (3.32) onto the family of Dirichlet Normal-Inverse-Gamma product distributions using the method of moments. We now prove the set of sufficient moments for the Dirichlet Normal-Inverse-Gamma product distribution is $\mathbb{S} = \{\mu_i, \sigma_i^2, \sigma_i^4, \mu_i^2\sigma_i^2, w_i, w_i^2\}_{i=1}^k$.

From (3.26), it is evident that $\boldsymbol{w}$ drawn from the Dirichlet distribution and $\Phi_i = \{\mu_i, \sigma_i\}$ drawn from the Normal-Inverse-Gamma distributions are independent random variables. Therefore, to find the sufficient moments of the Dirichlet Normal-Inverse-Gamma product distribution, we may use the sufficient moments for the Dirichlet and Normal-Inverse-Gamma distributions. The sufficient moments for the Dirichlet distribution are $\{\boldsymbol{w}, \boldsymbol{w}^2\}$ [124] and the sufficient moments of the i-th Normal-Inverse-Gamma distribution are $\{\mu_i, \sigma_i^2, \sigma_i^4, \mu_i^2\sigma_i^2\}$ [15, Chapter 4.5]. The concatenation of these sufficient moments then constitute the sufficient moments of (3.26).

The parameters of the Dirichlet Normal-Inverse-Gamma product distribution are then computed via (3.19)-(3.24) using the sufficient statistics as derived in [124] and [15], respectively. □

Theorem 5 is applied sequentially when multiple measurements are given. We summarize this procedure in Algorithm 4. The posterior predictive distribution is then given as $p(\psi|\hat{\Theta})$, which is a Gaussian mixture, where $\hat{\Theta} = \mathbb{E}[p(\Theta|\hat{\Psi})]$.

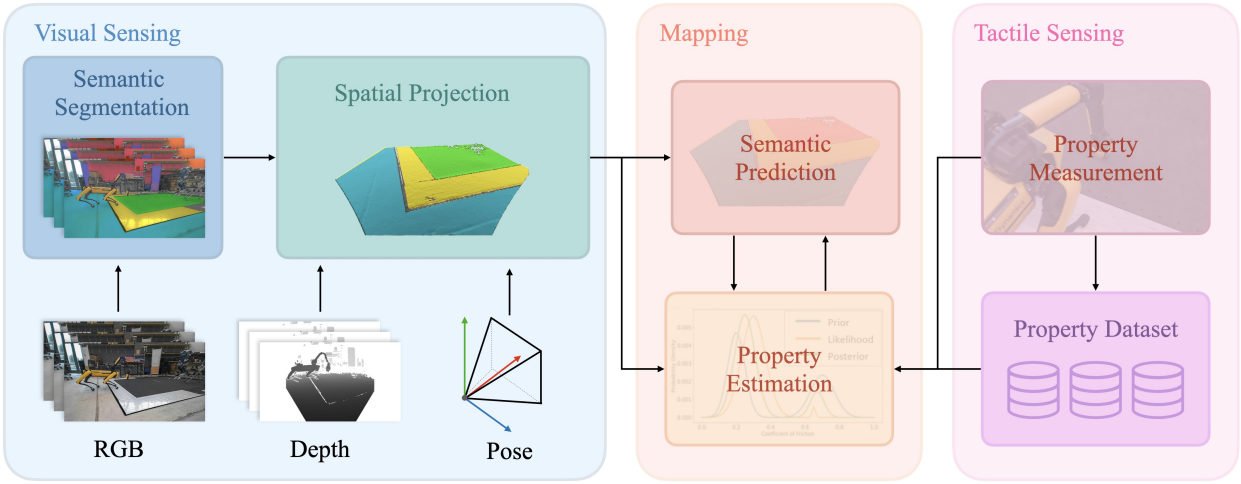


Figure 3.2: A flow diagram illustrating Algorithm 5. A semantic classification algorithm predicts pixel-wise classes from RGB images that are then projected into a common mapping frame using the aligned depth image, camera intrinsics, and estimated camera pose. This semantic point cloud is used to build a metric-semantic map. When a property measurement is taken, Algorithm 4 is used to update the semantic and property estimates.

3.3 Algorithm Overview

The proposed algorithm takes as input a stream of RGB-D images and physical property measurements from a tactile sensor and outputs semantic classifications and physical property estimates which are modelled probabilistically. We provide an overview of the proposed method presented in Algorithm 5. We focus on tactile and visual modalities, however our method may be adapted to different sensing modalities in a straight-forward manner.

A metric-semantic map and Dirichlet parameters are first initialized, where $\boldsymbol{\alpha} = \mathbf{1}_{k\times 1}$ (Lines 1 - 2). We assume that, while the robot is running, it is collecting exteroceptive data in the form of RGB-D images or lidar data. Using an off-the-self semantic segmentation network of choice, this data is transformed into a semantic point cloud [131] (Line 4). The geometric information of this point cloud is used to update the metric map (Line 5) while the semantic classifications are used to compute the semantic class posterior (Line 6), as described in Section 3.4.

Locations for property measurements are specified by the user. When a property measurement is taken using a tactile sensor, we first compute the region over which the measurement was taken within the map and use moment matching (Algorithm 4) to update the semantic classifications and property estimates within this region (Lines 8-10). We describe this step in more detail in Section 3.5. A flow diagram illustrating this procedure is given in Figure 3.2.

Algorithm 5: Semantic and Property Prediction

$\mathcal{M} \leftarrow$ initMap()
$\Theta \leftarrow$ initUncertainty()
while robot is running **do**
    $s \leftarrow$ getSemanticPointCloud()
    $\mathcal{M} \leftarrow$ updateMap(s) `// Sec.3.4`
    $\Theta \leftarrow$ updateSemanticUncertainty(s) `// Sec.3.4`
    if user requests property measurement taken **then**
        $\psi \leftarrow$ getMeasurement()
        $\mathcal{V} \leftarrow$ getMeasurementLocation() `// Sec.3.8`
        $\Theta \leftarrow$ measurementUpdate($\psi, \mathcal{V}$) `// Alg.4`

3.4 Vision-Based Estimation

This section describes how to implement Bayesian inference to recursively estimate semantic classifications using visual data. This corresponds to the *Visual Sensing* block in Figure 3.2. First, we discuss how to obtain a semantic segmentation point cloud and then discuss how to probabilistically fuse this data within a geometric representation.

Semantic segmentation assigns task-dependent class probability scores to each pixel in an image. It is commonplace to use neural networks to estimate pixel-wise class probability scores [63]. Let $I \in \mathbb{R}^{w \times h \times 3}$ denote an RGB image, where h, w are the height and width of the image in pixels. A trained semantic segmentation network takes an input image I and outputs the pixel-wise class probability scores $T \in \mathbb{R}^{w \times h \times k}$ for k semantic classes in the form of a k-dimensional Categorical distribution. By employing a one-hot encoding over these pixel-wise scores, $H = 1\{T\}$ such that $H \in \mathbb{R}^{w \times h}$, we may interpret the network as drawing pixel-wise class samples from pixel-wise Categorical distributions. We denote these as the *semantic class measurements*, the accuracy of which depends on the network used.

We use an aligned depth image, camera intrinsics, and camera pose estimate to project the semantic class measurements H into a fixed global coordinate frame [131, (10.38)] to obtain a point cloud representation of the semantically segmented image. Akin to [203] we implement a modified version of Kinect Fusion [88], a truncated signed distance field (TSDF), as the geometric representation onto which semantic class measurements are projected. Within each voxel we store unique Dirichlet parameters $\boldsymbol{\alpha}$. Given new semantic class measurements, we update these parameters in each voxel using (3.6). Thus, semantic classification estimates may be recursively obtained with uncertainty estimates via (3.5).

3.5 Tactile-Based Estimation

This section describes how semantic class estimates may be used to predict physical properties and how we use measurements of these properties to update semantic class likelihoods. This corresponds to the *Tactile Sensing* block in Figure 3.2. Our goal is to compute a measurement likelihood for physical properties using semantic class likelihoods introduced in Section 3.4.

Motivated by [47, 129], we construct a conditional probability distribution $p(\boldsymbol{\psi} \mid \mathcal{Z})$ for a property ψ with semantic data $\mathcal{Z}$. By applying the law of total probability and including the $\boldsymbol{\alpha}$ parameters introduced in Section 3.2.1.1, we derive the following expression:

$$p(\boldsymbol{\psi} \mid \mathcal{Z}, \boldsymbol{\alpha}) = \sum_{i=1}^{k} p(\boldsymbol{\psi} \mid z = i) p(z = i \mid \mathcal{Z}, \boldsymbol{\alpha}). \tag{3.33}$$

There are two components to this equation. The first, $p(z = i \mid \mathcal{Z}, \boldsymbol{\alpha})$, is the posterior predictive likelihood of a region of the map belonging to semantic class i given prior semantic class measurements within that region. This posterior predictive likelihood was derived in Section 3.2.1 and is given by (3.14).

Physical properties are not constant across a semantic class and thus should be estimated via a distribution of possible values, not by a single value. The second component, $p(\boldsymbol{\psi} \mid z = i)$, denotes this likelihood distribution and is approximated as a Gaussian distribution [47]. Substituting (3.14) and the formula for the class-wise Gaussian model into (3.33) gives a closed-form estimate for the physical properties given semantic class measurements:

$$p(\boldsymbol{\psi} \mid \mathcal{Z}, \boldsymbol{\alpha}) = \sum_{i=1}^{k} \frac{\alpha_i}{\sum_{j=1}^{k} \alpha_j} \mathcal{N}(\mu_i, \sigma_i^2). \tag{3.34}$$

Note that this model is equivalent to the Gaussian mixture model presented in (3.15) where the weights are computed using the predictive posterior of the semantic classifications. We use this as the measurement likelihood for the properties of interest.

Given that the Gaussian mixture model is used as the measurement likelihood, we use the approximate conjugate prior of this distribution, the Dirichlet Normal-Inverse-Gamma product, to represent uncertainty in the semantic classifications, w_i, and class-wise property parameters, μ_i and σ_i^2. Note that the Gaussian mixture weights, w_i, are computed using the Dirichlet parameters $\boldsymbol{\alpha}$. After a property measurement is taken, Algorithm 4 is implemented to update the belief over these parameters via (3.26). This algorithm is applied sequentially if multiple measurements are taken.



Figure 3.3: Results for a single simulated experiment. The image (a) and ground truth semantic labels (b) are from the Dense Material Segmentation Dataset [186]. The semantic segmentation predictions (c) do not classify parts of the desk as wood. A friction measurement is simulated using the ground-truth semantic label and Table 2.1 and is sampled from the pixel highlighted by the red cross in the RGB image. Algorithm 4 then computes the correct posterior semantic label (d).

Using Algorithm 4, the semantic class likelihood, w_i, as well as the property parameters themselves, μ_i and σ_i^2, are updated. Furthermore, the Dirichlet Normal-Inverse-Gamma product posterior provides a means of uncertainty quantification in the posterior estimates of the semantic class likelihoods and property values, unlike learning-based approaches.

3.6 Implementation

Algorithms 4 and 5 are implemented in C++ and include a ROS interface. Experiments are conducted on a laptop with a 5.4GHz i9-13900HX processor, 32GB of RAM, and an Nvidia RTX 4060 GPU.

We use a custom implementation of the SegFormer network [205] trained on the Dense Material Segmentation Dataset [186]. The output of the network is then post-processed with a segment-based voting scheme using FastSAM [55] in a similar manner to [28]. When a property measurement is taken, Algorithm 4 updates a region of the geometric representation segmented using FastSAM. This incentivizes regions with spatial proximity and visual similarity to be updated using a single measurement rather than requiring one measurement for each voxel, making the algorithm more efficient.

Table 2.1 is used to initialize the Gaussian parameters for the coefficient of friction for the relevant semantic classes. The values in Table 2.1 were computed using the material friction dataset provided by [47]. If a friction measurement is taken of a material class not present in Table 2.1, Algorithm 4 predicts the class from Table 2.1 with the most similar friction values.

For hardware demonstrations, static friction measurements are taken using a force-torque sensor which measures contact forces during the motion of an end-effector against a surface.

These forces are converted to friction measurements using the equation relating normal and tangential forces:

$$\psi = \frac{F_t}{F_n}, \tag{3.35}$$

where F_t is the force tangential to the contact normal, F_n is the force parallel to the contact normal, and ψ is the coefficient of friction. A low-pass filter is used to filter out measurement noise from the force-torque sensor.

3.7 Simulation Results

We validate our approach in simulation and demonstrate property measurements improve semantic predictions. The simulation experiments also validate the approximation accuracy of the moment matching method in Algorithm 4 in approximating the posterior via Theorem 5.

We use 1800 images and ground-truth semantic labels taken from the testing set of the Dense Material Segmentation Dataset [186] and the pre-trained semantic segmentation neural network described in Section 3.6 is used to predict semantic labels. Misclassified pixels in each image are randomly selected and a friction measurement is drawn from a Gaussian distribution whose parameters are specified in Table 2.1 using the corresponding ground-truth semantic label. The semantic prediction is used to initialize the Dirichlet parameters such that $\boldsymbol{\alpha} = 1_{k\times 1} + 1\{z = j\}$. The Gaussian parameters are initialized via Table 2.1 and the remaining Dirichlet Normal-Inverse-Gamma parameters are initialized as $\boldsymbol{a} = \boldsymbol{\alpha}$, and $\tau_i = \mu_i$, $\kappa_i = 1$, $\beta_i = \sigma_i^2/\tau_i$, $\gamma_i = \sqrt{\beta_i}/C$, and $C = 40$ for each semantic class.

After a simulated friction measurement is randomly sampled, Algorithm 4 is invoked and the posterior predictive likelihood is computed. The Segment Anything Model [28] computes a mask over the input RGB image using the chosen pixel as the query point and the semantic predictions over this region are updated corresponding to the posterior predictive likelihood output by Algorithm 4. An example of this computed posterior is demonstrated in Figure 3.3.

The mean pixel-wise accuracy of the semantic segmentation network over this dataset without applying the proposed algorithm is 27.2%. The mean pixel accuracy of the posterior semantic predictions output by Algorithm 4 is 33.0%, an increase of 21.3%, demonstrating increased semantic prediction performance compared to semantic predictions using vision alone.

Table 3.1 provides a further breakdown of the increase in the pixel-wise semantic prediction accuracy for each material class. Notably, materials such as rock, plastic, and concrete

Table 3.1: Increase in mean pixel-wise semantic prediction accuracy using simulated property measurements and Algorithm 4. Property measurements are shown to improve semantic prediction accuracy for all material classes from Table 2.1 except snow. The standard deviation for mean pixel-wise accuracy increase is also shown in the second column. The third column shows the maximum increase in pixel-wise semantic prediction accuracy seen in the simulation experiments.

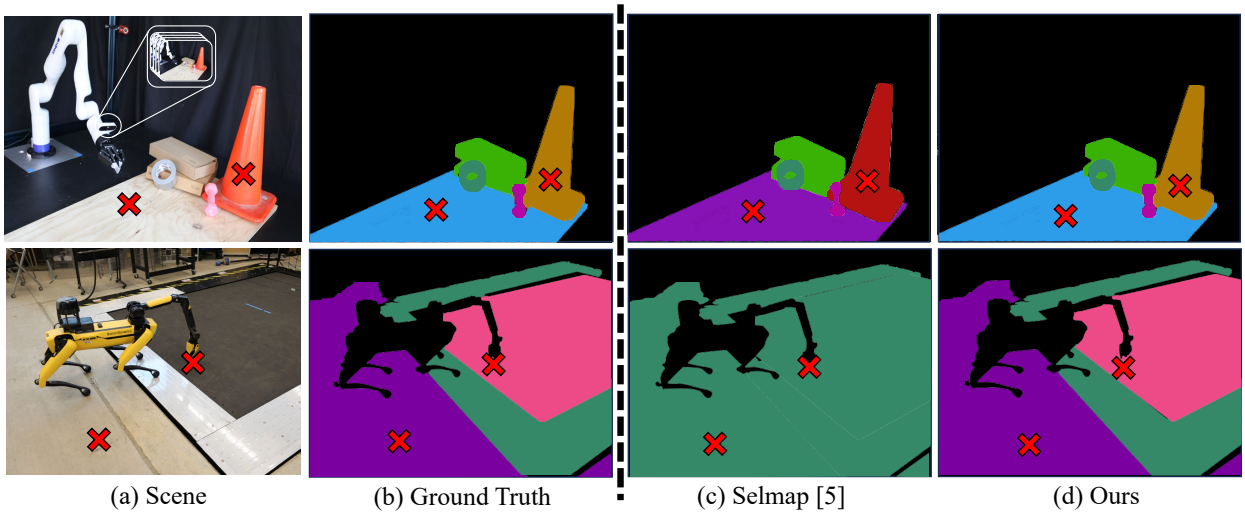
Semantic Class	mAcc % Increase	Max Acc % Increase
Concrete	38.6 ± 51.9	192.8
Grass	14.2 ± 32.8	129.8
Rock	26.4 ± 41.1	191.2
Wood	24.5 ± 35.0	190.0
Rubber	5.7 ± 12.3	50.8
Plastic	13.5 ± 26.6	184.4
Snow	0.00 ± 0.00	0.00
Ice	110.1 ± 0.0	110.1
Total	21.3 ± 34.3	192.8

occupy the majority of pixels in some images, which leads to increases in pixel-wise accuracy of over 75%. All classes except for snow saw improved semantic prediction performance using simulated tactile sensing. When simulating tactile measurements for snow, Algorithm 4 incorrectly predicts the semantic class as either rock or wood because the Gaussian parameters for snow are similar to those of these two classes. Interestingly, depending on the simulated friction measurement, the Gaussian mixture mode responsible for predicting snow collapses and the mode for either rock or wood becomes dominant. Only one image in the Dense Material Segmentation Dataset test set contained ice.

This experiment highlights that by conditioning semantic classifications on physical properties, one can generally improve the accuracy of visual semantic classifications.

3.8 Hardware Experiments

Next, we validate our method on several hardware demonstrations and compare against existing semantic mapping and property estimation approaches. In the first demonstration, we validate the results from simulation and show that our approach is capable of correcting erroneous semantic classifications by measuring the coefficient of friction. We then provide two case studies for the utility of estimating physical properties. For the first case study, our proposed approach is used within a risk-aware legged locomotion planner to update the locomotion gait of a quadruped when hazardous terrain is perceived. In the second case study, we demonstrate our approach is able to estimate affordance-based properties



(a) Scene (b) Ground Truth (c) Selmap [5] (d) Ours

Figure 3.4: A semantic segmentation task is shown and the proposed method is compared against the semantic mapping approach from [47] which is called Selmap. The expected semantic class is shown. a) The input image with measurement locations shown using an X and associated b) ground truth semantic labels are provided. Both Selmap and our approach use the same SegFormer + FastSAM pre-trained semantic segmentation network specified in Section 3.6. This network incorrectly predicts the semantic labels for several regions within the scene. c) This incorrect labelling is reflected in the Selmap implementation as the visual-based semantic mapping approach is unable to correct these erroneous predictions. By exploiting a tactile sensing modality, d) our approach is able to correct the erroneous semantic predictions and correctly predict the semantic labels of the objects within the scene.

in a door-opening scenario. Both case studies are presented on the accompanying project webpage.

3.8.1 Semantic Prediction Using Property Measurements

The end-effector of a Kinova Gen 3 robotic arm and a Spot Arm are used to collect friction measurements using built-in wrist-mounted force-torque sensors. For this experiment, 5 frames from the RGB-D stream are used to initialize the semantic TSDF map. Each RGB-D image is semantically segmented using the trained network and projected into the global coordinate frame using the aligned depth image. The recursive vision-based semantic update described in Section 3.4 is applied for each semantic point cloud. This initializes the $\boldsymbol{\alpha}$ parameters in (3.34) used to compute the initial semantic classification weights for the measurement likelihood. The Gaussian parameters are initialized via Table 2.1 and the remaining Dirichlet Normal-Inverse-Gamma parameters are initialized as per Section 3.7.

We compare our approach to the recursive semantic mapping approach of [47] which only uses vision. The same SegFormer-FastSAM semantic segmentation network is used for both

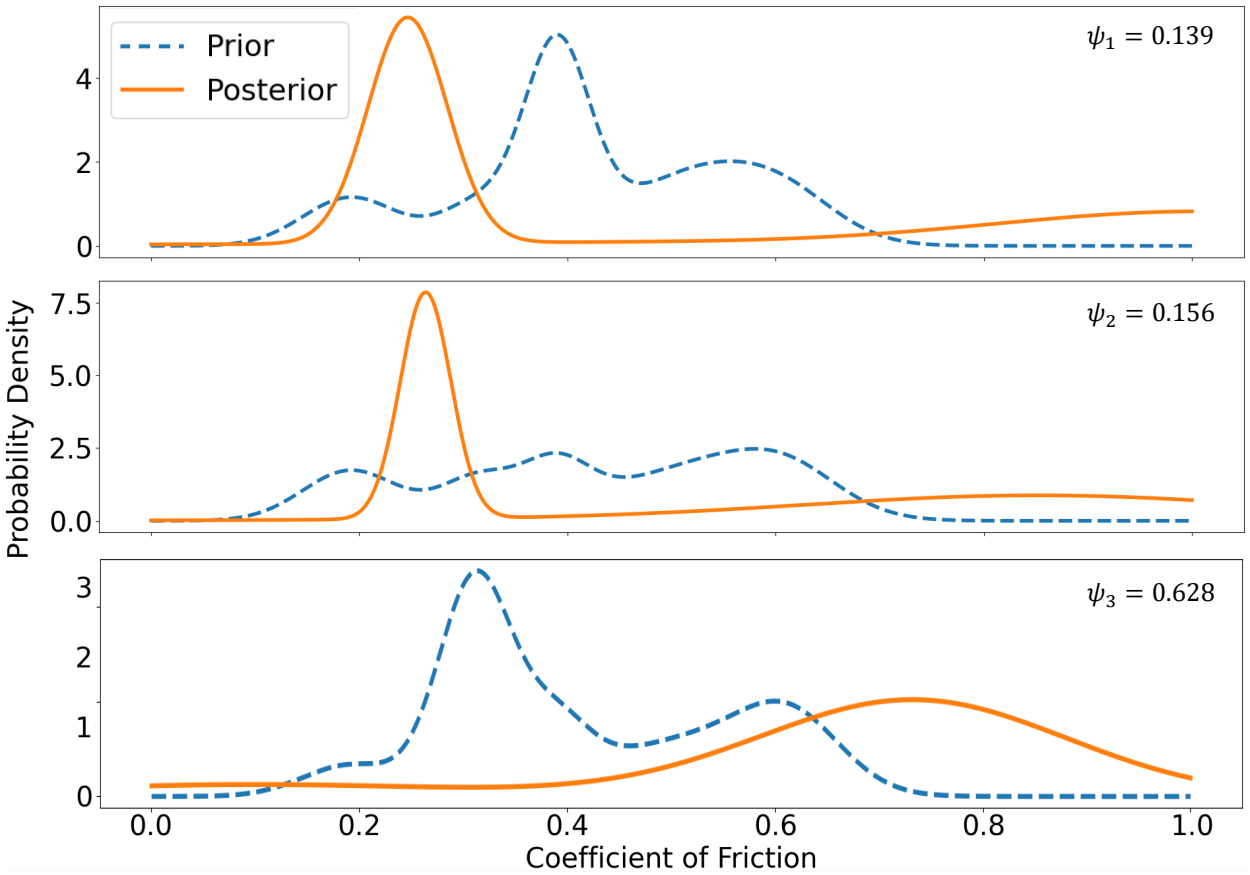


Figure 3.5: The prior (dotted) and posterior (solid) friction estimates for the three ice traversal demonstration runs. When friction measurements are taken of the ice before traversal, Algorithm 4 predicts the posterior friction estimates. The friction measurements are $\psi_1 = 0.139$, $\psi_2 = 0.156$, and $\psi_3 = 0.628$ for trials 1, 2, and 3, respectively.

methods. As shown in Figure 3.4, the network incorrectly classifies various objects in the scenes. When only vision-based semantic classifications are considered [47], the erroneous predictions from the network are unable to be corrected (Figure 3.4c).

Using the proposed method, a user specifies the location for friction measurements to be taken (Figure 3.4a). Algorithm 4 then computes the posterior semantic classification weights using the friction measurements as input. This corrects the expected semantic classification, as shown in Figure 3.4d.

We ran the experiment on two indoor scenes with two friction measurements each as depicted in Figure 3.4. As shown in Figure 3.4, the semantic prediction accuracy increases using the proposed method and matches the ground-truth semantic labels. In contrast, [47] is unable to correct the erroneous predictions from the semantic segmentation network, resulting in incorrect semantic predictions. This demonstrates that the simulation results

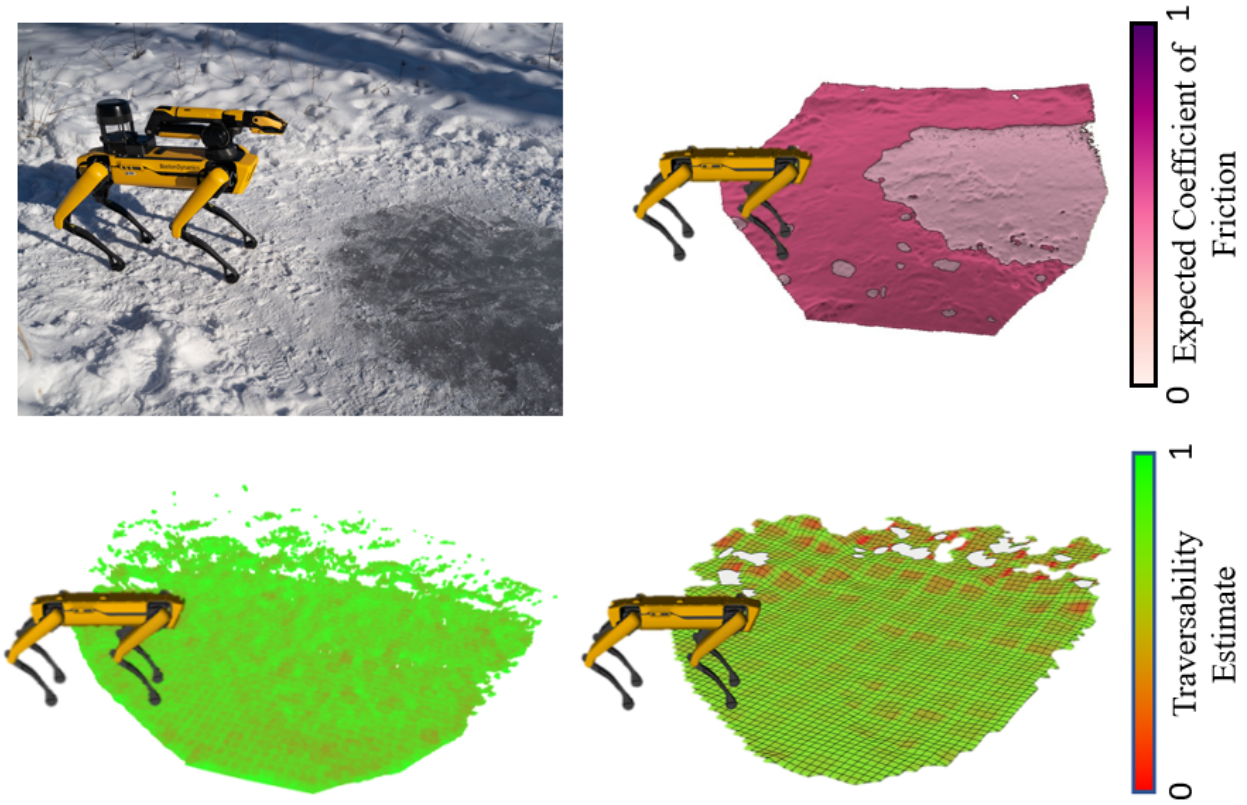


Figure 3.6: The performance of the proposed approach after a friction measurement is taken in comparison to the traversability estimate computed by [2] (bottom left) and [61] (bottom right). The proposed method is able to correctly predict that the surface in front of the robot has a low coefficient of friction which informs the locomotion planner to switch to a stable gait. In contrast, the traversability estimates predict the ice is traversable even when the robot is using a dynamic gait, which results in the robot slipping and falling.

shown in Section 3.7 translate to hardware experiments.

3.8.2 Friction Estimation Using Semantics

We demonstrate the utility of our proposed property estimation method using a case study involving a challenging legged locomotion traversal task of crossing an icy surface. We implement a locomotion planner similar to [51] that switches between predetermined static and dynamic gaits depending on the expected value for the coefficient of friction of the terrain to be traversed. A static gait is used when the expected value for the coefficient of friction falls below a threshold of $\mathbb{E}[\psi] \leq 0.25$.

A stream of RGB-D images is used to build a semantic TSDF map of the robot's environment as outlined in Section 3.4. The resultant semantic TSDF map predicts that the region the robot must traverse may have a low coefficient of friction and a measurement is

therefore taken to better approximate the value.

The Dirichlet Normal-Inverse-Gamma prior parameters are initialized as described in Section 3.7. We use the manipulator mounted to the legged robot to take friction measurements of the ground in front of the robot using a built-in wrist-mounted force-torque sensor via (3.35). Algorithm 4 then computes the posterior estimate for the coefficient of friction (Figure 3.1).

The experiment was conducted six times in the same location during the same afternoon, three where the robot does not measure the coefficient of friction and three where it does. The prior and posterior friction estimates for these six trials is shown in Figure 3.5. The semantic segmentation network incorrectly predicts the semantic label of the terrain for each run and mis-classifies the ice as either concrete or snow. Since a friction measurement is not taken, the friction threshold $\mathbb{E}[\psi] \leq 0.25$ is not met due to mislabeling and the robot attempts to cross the ice using a dynamic gait, resulting in it slipping and falling for all three of these trials.

For the subsequent two trials, friction measurements of $\psi_1 = 0.139$ and $\psi_2 = 0.156$ are taken for each trial, respectively. After computing the posterior friction estimates via Algorithm 4, the friction threshold is met and the robot switches to a static, stable gait to cross the ice. This results in successful ice traversal for these two trials. In the following third trial, a friction measurement of $\psi_3 = 0.628$ is taken due to erroneous end-effector placement. The computed posterior is shown in Figure 3.5 and shows a coefficient of friction with a large expected value which exceeds the threshold. For this trial, the robot remains in a dynamic gait and slips on the ice during traversal. A video of a successful ice traversal trial is provided on the accompanying project webpage.

We further compare our property estimation approach with two state-of-the-art traversability estimation methods designed for legged locomotion. Both methods consider both geometric and semantic features and use a pre-trained neural network to estimate a traversability score [2, 61]. Both traversability estimation methods predict that the icy surface is traversable as shown in Figure 3.6; however, attempting to traverse this surface using a dynamic gait resulted in the robot slipping and falling.

Our proposed method is able to verify the property predictions estimated using visual information via tactile sensing and Bayesian inference whereas other state-of-the-art property estimation methods are unable to incorporate multiple sensing modalities. This task highlights the utility of estimating physical properties such as the coefficient of friction to accomplish challenging tasks such as legged locomotion traversal across an icy surface. State-of-the-art traversability estimation methods depend only on the terrain geometry and classification and thus do not provide the means by which a robot may adapt its traversal

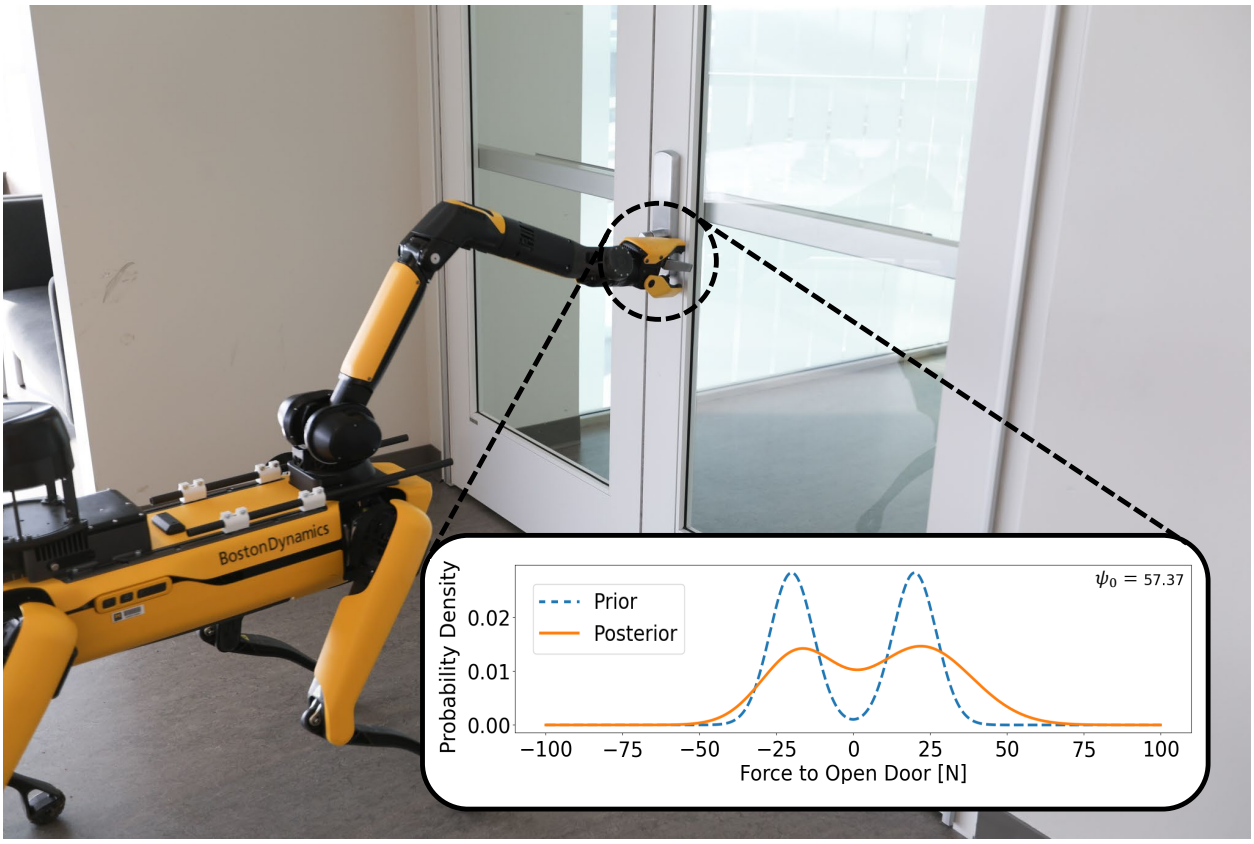


Figure 3.7: The proposed method is capable of encoding affordance-based properties, in this case the force required to open a door. This door opening force is initialized as a multi-modal distribution with pushing and pulling being equally likely with a magnitude of 20N. The robot applies a pulling force of 57N on the door, resulting in it swinging open. The posterior estimate for the force required to open a door is computed and shown in the embedded plot.

strategy to account for the properties (i.e. friction) which dictate traversability.

3.8.3 Encoding Affordance-Based Properties

Lastly, we present a scenario in which our approach is used to estimate affordance-based properties. An affordance is a property of an object that defines its possible use [69]. In this case, we focus on the affordance of a door handle, which affords either a push or pull action, as the semantic classification and examine the force required to open the door as the property of interest.

As shown in Figure 3.7, a mobile manipulator is used to measure the force required to open a door. Negative forces constitute pushing while positive forces constitute pulling. We equally weight each push/pull affordance and initialize a mean force magnitude of $20N$ and

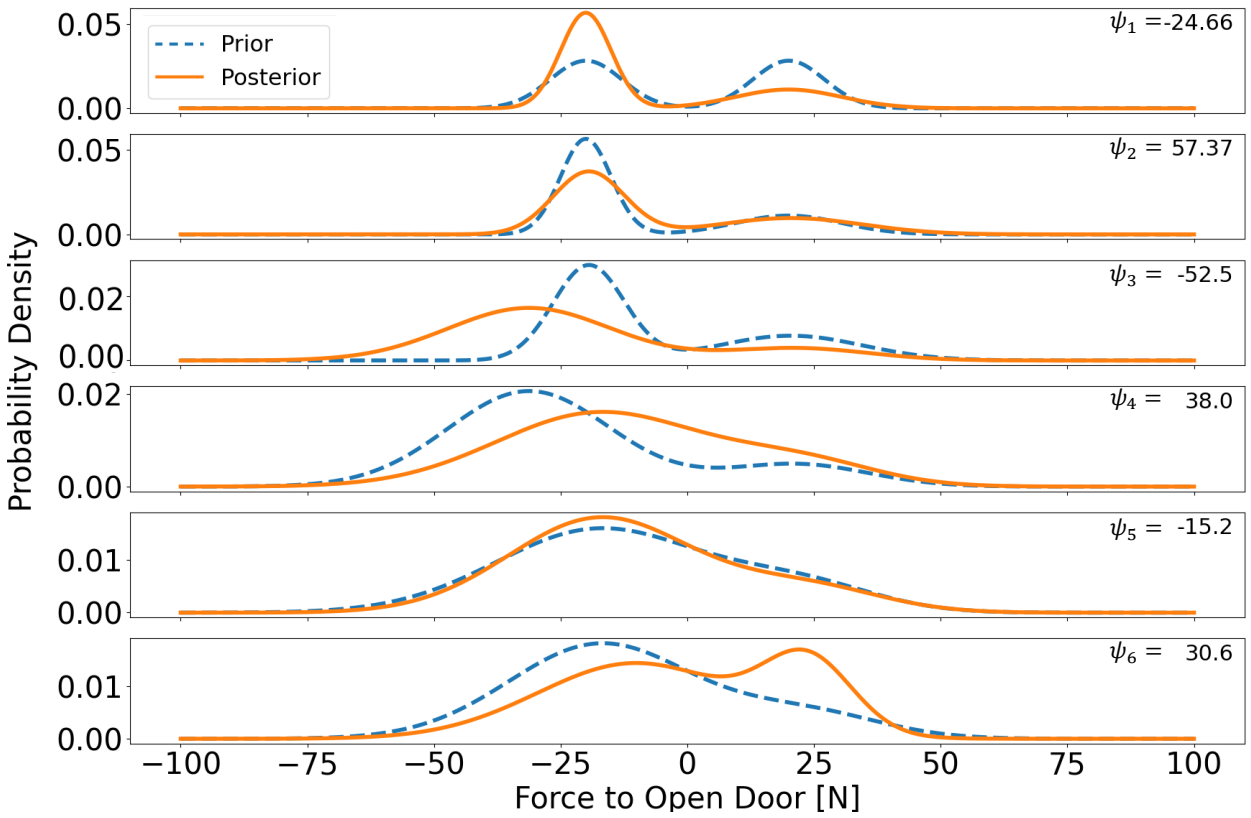


Figure 3.8: The prior and posterior door opening force estimates running Algorithm 4 sequentially with six experiments, meaning the posterior for the previous experiment is the prior for the current experiment. The force required to open the door is indicated in the upper right-hand corner for each experiment. After measuring the force required to open 6 doors, the posterior estimate has a high variance with an expected value near $0N$, reflecting the equal likelihood that a door is either push or pull to open.

a variance of $10N$.

To measure the force required to open the door, the robot begins applying a pulling force $10N$ and ramps this up to $70N$ over a period of 3 seconds. At $57N$ the door begins to open and this value is used as the property measurement for Algorithm 4. The door opening force posterior is show in Figure 3.7 in the lower right-hand corner. After invoking Algorithm 4 the push/pull affordances are still almost equally weighed. Additionally, the variance for the pulling force mode has increased, mirroring the fact that a higher-than-expected force was required to open the door. This door opening experiment is demonstrated on the accompanying project webpage.

Next, we test the force required to open six different doors, three pull-to-open and three push-to-open, and use Algorithm 4 to compute the posterior door opening force estimate sequentially after each trail. Figure 3.8 shows the prior and posterior opening force estimate

after each trial. After six trials the estimated door opening force posterior has maintained two modes corresponding to the push or pull force modalities, however the variance has increased, representing the varied door opening forces seen in the experiments compared to the initial estimate.

This case study demonstrates the utility of the proposed method for affordance-based property estimation and opens an avenue for future research given recent advances in affordance-based planning [152].

3.9 Discussion and Conclusion

We propose a method for jointly estimating semantic labels and physical properties. We model semantic classifications and physical properties probabilistically enabling closed-form Bayesian inference given visual semantic predictions and tactile property measurements. We demonstrate that by leveraging this probabilistic approach we are able to improve semantic prediction accuracy by an average of 21.3% in simulation. Additionally, we provide two case studies which motivate the utility of physical property estimation demonstrated in the proposed method.

One drawback of our approach is that the accuracy of our method is reliant on the accuracy of the semantic classification network. If a semantic classification network consistently predicts incorrect semantic labels over a long duration, the initial Dirichlet parameters will skew heavily towards the incorrect semantic label and be difficult to correct with property measurements. Additionally, like many voxel-based semantic mapping frameworks, our method is constrained by memory limitations which restrict the size of the map. We aim to address these concerns in future work. Tactile and visual sensing modalities were used for the proposed method, however additional sensing modalities may also be used in place of or in addition to tactile sensing. This is an avenue for future work. Lastly, our presented method provides a link between visual and tactile sensing modalities through physical property estimates and future work will aim to exploit this relationship for active learning.

CHAPTER 4

Uncertainty Quantification for Radiance Field Models

Radiance field models have become fundamental for novel view synthesis by enabling the generation of photo-realistic images from different perspectives based on sparse input views [150]. These models excel at learning continuous volumetric scene representations, enabling high-quality, view-consistent rendering with utility in applications such as computer vision [148] and robotics [121]. While radiance field models have seen widespread use, a notable challenge remains in quantifying the uncertainty of these models. Consequently, it can be challenging to determine when these models yield accurate approximations of scenes, detracting from their robustness and reliability, especially for applications requiring high levels of confidence in the generated views.

Uncertainty quantification in radiance field models offers valuable insights into regions of the environment that the model does not represent well and areas where additional data could improve model accuracy. This uncertainty can then be leveraged for downstream applications such as active perception [92] or safety [121].

Existing uncertainty quantification approaches for radiance field models attempt to model uncertainty in the model parameters directly and approximate model output uncertainty in the form of rendered uncertainty [174, 70]. Unfortunately, the non-linearity of the rendering function means computing pixel-wise probability densities from model parameter uncertainty is computationally intractable. Existing approaches therefore rely on approximations or heuristics to transform model parameter uncertainty into rendering uncertainty, sometimes training neural networks to estimate uncertainty directly [92, 164], approximating uncertainty over a 3D deformation field in a post-hoc manner [70], or relying on Gaussian approximations [91]. However, the relationship between these uncertainty approximations and rendering error remains untested. Furthermore, the complexity of these approximations often results in increased computational costs.

To overcome these challenges, we present a framework for quantifying uncertainty of radi-

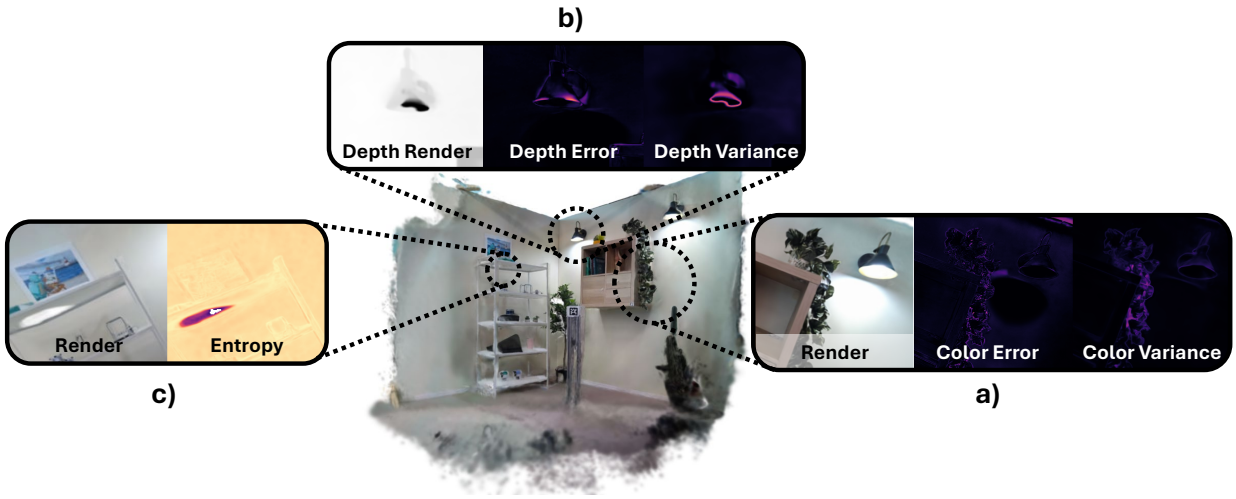


Figure 4.1: The ground-truth, rendered image, color variance, and color error for the Mip360 dataset. These regions of high variance correspond to regions of high rendering error, as demonstrated in Tables 4.1. Regions of high variance also correspond to regions with many translucent Gaussians and complex texture, such as grass and vegetation, indicating these regions of the image have not converged.

ance field models outputs directly. Examining the rendering equation through a probabilistic lens, we demonstrate the rendering output can be modeled as a probability distribution for which the moments and entropy can be easily computed with minimal additional computational cost. We demonstrate the efficacy of our approach by first showing that the pixel-wise variance rendered using our method has a higher correlation with rendering error than existing radiance field uncertainty quantification methods. Next, we demonstrate that this model output uncertainty is a better metric for next-best-view selection compared to other post-hoc methods.

In summary, the contributions of this chapter are:

- A means of computing the higher-order moments of the pixel-wise uncertainty and entropy derived from the radiance field rendering equation,
- Demonstrate the pixel-wise variance is strongly correlated to rendering error,
- Demonstrate pixel-wise uncertainty is a better signal for next-best-view selection than existing uncertainty approximation approaches.

The remainder of the chapter is organized as follows. Section 4.1 presents an overview of uncertainty quantification in radiance field methods from the literature. Section 4.2 describes how we model rendering as a probabilistic process and how we derive the higher-order moments and entropy of the associated render. Section 4.3 demonstrates the efficacy of our



Figure 4.2: The ground-truth, rendered image, color variance, and color error for the Mip360 dataset. These regions of high variance correspond to regions of high rendering error, as demonstrated in Tables 4.1. Regions of high variance also correspond to regions with many translucent Gaussians and complex texture, such as grass and vegetation, indicating these regions of the image have not converged.

approach using three case studies: correlation with rendering error and next-best-view selection. Finally, Section 4.4 provides a discussion and concluding remarks.

4.1 Related Works

Uncertainty quantification plays a critical role in computer vision for enhancing model reliability and estimating prediction error [176]. This includes estimating both aleatoric uncertainty, uncertainty in the data, and epistemic uncertainty, uncertainty in the prediction itself [3].

Network dropout [58] and Bayesian neural networks (BNNs) [102, 126] are two well-established techniques used to quantify uncertainty in computer vision tasks using Monte Carlo and functional approximations, respectively. While dropout has proven effective and offers a balance between computation cost and accuracy [169], BNNs can be computationally intensive to train and may not generalize to under domain shift [89]. Ensemble methods also approximate uncertainty, however training a sufficient number of models is computationally intensive and inference time scales with the number of models [174].

Radiance field models have seen a surge in popularity in recent years spurred by the

development of neural radiance fields (NeRF) which uses a neural network to approximate the 5-dimensional radiance field function [123]. Recently, explicit models using Gaussian basis functions to approximate the radiance field have gained popularity, namely 3D Gaussian Splatting (3DGS) [94]. These models trace their have foundations to work in texture mapping [79], however they have shown state-of-the-art performance in novel view synthesis and scene representation [27].

Perspectives on how to estimate the uncertainty of radiance field models remain varied. For NeRF-based models, dropout and ensemble [174] approaches remain popular, however ensemble methods are computationally intensive to compute. Other prominent uncertainty quantification methods include S-NeRF [165] and CF-NeRF [164], which model uncertainty over the set of possible radiance field models. Evidential regression [3] has also shown promise for uncertainty quantification by training a network to output an additional variance parameter [92, 164], although fundamental concerns about this approach have been raises [202, 116]. In contrast, post-hoc approaches which approximate the volumetric uncertainty of a NeRF model without modifying the training process have also been proposed [70]. Notably, it was found approximating uncertainty of the NeRF network parameters directly was infeasible due to high correlation between the MLP layers.

Parallel uncertainty quantification methods for 3DGS models have recently been proposed. Variational inference is used as a training signal in [158] to predict the quality of the 3DGS model while FisherRF [91] first estimates Gaussian parameter uncertainty and renders a heuristic approximation of this uncertainty. Unlike NeRF-based approaches, it was found that the parameters in the 3DGS model have minimal correlation [91], and second-order Laplace estimates are used to capture this parametric uncertainty. Unfortunately, uncertainty approximation for the 3DGS model parameters fails in regions with many Gaussians, as the gradient of the render with respect to any single Gaussian is minimal. There are also methods which use heuristics for uncertainty, and leverage these heuristics for training and regularization [95, 84].

In contrast to these approaches, we present a general, unified approach for estimating the uncertainty in radiance field model outputs by considering the rendering equation as a probabilistic process. Our proposed approach works with both NeRF and 3DGS models and is able to render color, depth, and semantic uncertainty.

4.2 Methodology

4.2.1 Radiance Fields

A radiance field models the image formation process by approximating a five-dimensional function that maps from a point and viewing direction to the opacity $\sigma : \mathbb{R}^3 \to \mathbb{R}$ and the color $c : \mathbb{R}^3 \times S^2 \to \mathbb{R}^3$ along the viewing ray. Such radiance field models have also been extended to represent depth [33, 87] and semantics [148] alongside color. Intuitively, σ represents the likelihood that a particle traveling along a ray will stop when it collides with an individual point in space. Under a radiance field model, an image is rendered by considering separately the ray that intersects the camera origin and each pixel. Consider such a ray $r(s) : \mathbb{R} \to \mathbb{R}^3$. We define an additional quantity, the *transmittance*, which defines the unnormalized probability that a particle traveling along ray r from $s = a$ to $s = b$ will experience a collision. This transmittance, denoted as $\mathbb{T}_a^b[r]$, is computed by

$$\mathbb{T}_a^b[r] = \exp\left(-\int_a^b \sigma(r(s))ds\right) \tag{4.1}$$

Equivalently, $\mathbb{T}_a^b[r]$ may be interpreted by considering a light source aligned with ray r and located at $s = a$. Then $\mathbb{T}_a^b[r]$ is the proportion of light that remains by point $s = b$.

From the transmittance, the quantity of interest ρ, representing either color, depth, or semantics, of the pixel corresponding to ray r is computed as

$$I[\rho] = \int_{z_n}^{z_f} \rho(r(s))\sigma(r(s))T_{z_n}^s[r]ds \tag{4.2}$$

where z_n, z_f are the near and far clipping planes and $\sigma(r(s))$ represents the normalizing factor for the probability that a particle traveling along ray will experience a collision.

4.2.2 Gaussian Splatting Models

Gaussian splatting models represent radiance fields using 3D Gaussian basis functions. Each 3D Gaussian consists of a learned opacity, α, mean and covariance of the Gaussian, μ and Σ, as well as a color represented by spherical harmonics [94, Section 2.3]. Recent approaches also include semantic embeddings, φ. We combine all parameters together such that $\boldsymbol{w} = \{\alpha_j, \mu_j, \Sigma_j, c_j, \varphi_i\}_{j=1}^M$, where M is the number of Gaussians in the splatting model.

Unlike neural radiance fields, Monte Carlo integration along rays is not required for Gaussian Splatting as the 3D Gaussians are projected onto the image plane and subsequent pixel colors, depths, and semantics are computed. Given a camera pose $(\boldsymbol{x}, \boldsymbol{d}) \in SE(3)$, the

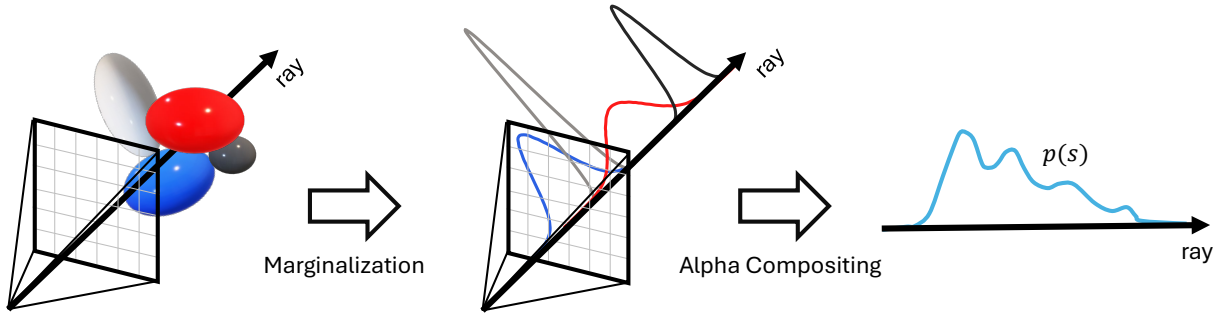


Figure 4.3: Pixel-wise higher-order moments and entropy are computed in the Gaussian Splatting rasterizer. The 3D Gaussian basis functions are first marginalized by projecting them onto the image plane and ordered based on distance. For each pixel, the alpha compositing process effectively computes the probability of ray termination, $p(s)$. This probability density is then used to compute the higher-order moments and entropy of each pixel.

rendering function (4.2) for the i-th pixel, ρ_i, in the image plane is

$$\rho_i(\boldsymbol{x}, \boldsymbol{d}) = \sum_{j \in \mathcal{J}} \rho_j \tilde{\alpha}_j(\boldsymbol{x}, \boldsymbol{d}) \prod_{k<j} (1 - \tilde{\alpha}_k(\boldsymbol{x}, \boldsymbol{d})) \tag{4.3}$$

where ρ_j is either the color, depth, or semantic embedding of the j-th ordered Gaussian and

$$\tilde{\alpha}_j(\boldsymbol{x}, \boldsymbol{d}) = \alpha_j \cdot \exp\left(-\frac{1}{2} \left\|\boldsymbol{x} - \boldsymbol{\mu}_j(\boldsymbol{d})\right\|^2_{\Sigma_j^{-1}(\boldsymbol{d})}\right) \tag{4.4}$$

is the pixel-wise opacity of the j-th ordered Gaussian in set $\mathcal{J}$. Here, the set $\mathcal{J}$ is constructed by sorting the projected Gaussians by depth with respect to the image plane, and $\boldsymbol{\mu}_j$ and Σ_j are the 2D mean and variance of the j-th splatted Gaussian. As discussed in [94], the α-blending in (4.3) saturates once $\sum_{j \in \mathcal{N}} \alpha_j = 1$ and all remaining Gaussians in the ordered set $\mathcal{N}$ are not considered. Notably, the ordering of the Gaussian splats is approximated since the rotation of each 3D Gaussian is neglected [94].

This splatting process may also be interpreted as marginalizing the Gaussian basis functions onto the ray creating a Gaussian Mixture Model (GMM). The weights of each mode in the GMM correspond to the opacities computed in the alpha compositing process (Fig. 4.3).

4.2.3 Uncertainty Quantification in Model Outputs

Our approach aims to quantify uncertainty in the image space rather than the parameter space of the Gaussian splat model. The rendering equation for the Gaussian splatting model

Table 4.1: Correlation coefficients for color variance and rendered color error.

Method	Synthetic NeRF			Mip360			TUM Dataset		
	τ_p	τ_s	τ_k	τ_p	τ_s	τ_k	τ_p	τ_s	τ_k
FisherRF	0.580	0.215	0.137	0.527	-0.007	-0.005	0.482	-0.071	-0.046
Ours	0.716	0.838	0.716	0.668	0.368	0.254	0.717	0.572	0.402

Table 4.2: Correlation coefficients for depth variance and rendered depth error.

Method	Synthetic NeRF			TUM Dataset		
	τ_p	τ_s	τ_k	τ_p	τ_s	τ_k
FisherRF	0.572	0.328	0.238	0.546	0.575	0.405
Bayes' Rays	0.758	0.473	0.364	-	-	-
Ours	0.866	0.947	0.833	0.671	0.685	0.519

(4.3) is nonlinear with respect to the parameters of the model, meaning computing the render likelihood directly from parameter uncertainty is computationally intractable. Our approach instead considers uncertainty in pixel space and uses this pixel-wise uncertainty to represent model uncertainty.

We note that the rendering equation is the expectation over the quantity of interest, $\rho(r(s))$:

$$\mathbb{E}[\rho] = \int_{z_n}^{z_f} \rho(r(s))\sigma(r(s))T^s_{z_n}[r]ds. \tag{4.5}$$

The random variable in the expectation is s, representing the event of a particle collision along the ray, and $\sigma(r(s))T^s_{z_n}[r]$ is the associated probability distribution of this collision. As the quantity of interest, ρ, is a function of this random variable, the expected value, moments, and entropy of ρ are therefore trivially computed [154]. From this probabilistic interpretation of the rendering equation, the j-th higher-order moment for each pixel is computed as:

$$\mathbb{E}[\rho^j] = \int_{z_n}^{z_f} \rho(r(s))^j\sigma(r(s))T^s_{z_n}[r]ds. \tag{4.6}$$

The variance of each pixel is computed using (4.5) and the second-order moment:

$$\mathrm{Var}[\rho] = \mathbb{E}[\rho^2] - \mathbb{E}[\rho]^2. \tag{4.7}$$

Furthermore, by considering the quantity $\sigma(r(s))T^s_{z_n}[r]$ as the probability of a particle

collision along the ray, the associated entropy of the ray is computed as:

$$H[\rho] = \int_{z_n}^{z_f} p(s) \log p(s) ds \tag{4.8}$$

$$= \int_{z_n}^{z_f} \sigma(r(s)) T^s_{z_n}[r] \log \left(\sigma(r(s)) T^s_{z_n}[r] \right) ds. \tag{4.9}$$

We note that moment and entropy computations make no assumptions on the form of the random variable. Furthermore, the entropy computation is not a function of the rendered quantity of interest. Rather, the entropy depends on the scale, orientation, and location of the Gaussian basis functions, demonstrating that the entropy of the Gaussian splatting renders are determined exclusively by the spatial properties of the 3D Gaussian basis functions.

For the 3DGS model, the higher-order moments are computed via:

$$\mathbb{E}[\rho^n] = \sum_{j \in \mathcal{J}} \rho_j^n \tilde{\alpha}_j(\boldsymbol{x}, \boldsymbol{d}) \prod_{k<j} (1 - \tilde{\alpha}_k(\boldsymbol{x}, \boldsymbol{d})), \tag{4.10}$$

and the entropy is computed as

$$H[\rho] = \sum_{j \in \mathcal{J}} p(s_j) \log(p(s_j)), \tag{4.11}$$

where

$$p(s_j) = \tilde{\alpha}_j(\boldsymbol{x}, \boldsymbol{d}) \prod_{k<j} (1 - \tilde{\alpha}_k(\boldsymbol{x}, \boldsymbol{d})). \tag{4.12}$$

In the following section we demonstrate that the variance renders are better correlated with the rendering error than existing uncertainty estimation approaches. We then demonstrate that the rendered uncertainty is a better selection criteria for next-best-view selection than existing uncertainty quantification approximations.

4.3 Experiments

In this section we demonstrate the utility of our uncertainty rendering approach across two case studies. First, we demonstrate that our variance rendering framework has a strong correlation with the rendering error. Next, we show that variance renders using our proposed approach provide a better next-best-view selection criteria than existing state-of-the-art methods.

Table 4.3: Next best view selection on the Synthetic NeRF dataset with 10 views and the TUM dataset with 200 views. The first, second, and third best are denoted in red, orange, and yellow, respectively.

Method	Synthetic NeRF			TUM Dataset		
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)
Random	24.009	0.907	0.079	19.175	0.703	0.382
FisherRF	24.733	0.911	0.073	18.978	0.700	0.388
Ours (Entropy)	24.211	0.909	0.078	19.093	0.701	0.383
Ours (Depth Var.)	24.577	0.913	0.073	18.946	0.697	0.388
Ours (Color Var.)	24.776	0.916	0.072	19.191	0.703	0.382

4.3.1 Error Correlation

We determine the relationship between our proposed variance rendering approach and the rendering error for both color and depth. A correlation between variance and error indicates that our approach correctly models uncertainty in the Gaussian splat and is able to determine regions of the scene for which the 3DGS model is a poor fit to the scene.

In this case study we compute the Pearson correlation coefficient τ_p [35], Spearman rank-order coefficient τ_s [160], and Kendall tau coefficient τ_k [161]. These coefficients each model the strength and direction of correlation using different metrics. A coefficient of 1 indicates a strong correlation, meaning regions of high rendered variance correspond to regions of high rendering error. A coefficient of -1 indicates a strong inverse correlation, and a coefficient of 0 indicates no correlation.

We compare our proposed variance rendering approach with Bayes' Rays and FisherRF, two state-of-the-art methods which approximate model output uncertainty as a function of model parameter uncertainty. The correlation coefficients for the Synthetic NeRF [123], Mip360 [8], and TUM [172] datasets for all methods are presented in Tables 4.1 and 4.2.

Notably, our proposed variance rendering method outperforms all baselines across all datasets for both color and depth correlation. This demonstrates that our variance rendering method better represents model uncertainty in regions of the scene for which the radiance field model is a poor representation. Such regions can then be queried for ground-truth information and this information subsequently used to train the radiance field model to improve model performance in these regions. We demonstrate such a use-case in the following subsection.

4.3.2 Next-Best-View Selection

The Next-Best-View (NBV) problem in computer vision refers to the task of determining the optimal subsequent viewpoint that maximizes the information gained, often in the context of 3D scene reconstruction, object recognition, or inspection tasks. This problem is crucial in scenarios where capturing a comprehensive model of an environment or object with minimal sensing actions is desired. Existing methods for NBV estimate the uncertainty or entropy of the model parameters directly [70, 91], and then approximate model output uncertainty as a function of parameter uncertainty. The viewpoint with the highest model parameter uncertainty is selected, thus improving scene completeness or reconstruction accuracy by providing more information about these parameters.

We demonstrate that computing model output uncertainty directly is a stronger selection criteria for NBV selection by comparing this approach to state-of-the-art NBV methods for radiance fields. We frame NBV as a Bayesian optimization problem [53] using the Gaussian Process Upper Confidence Bound acquisition function with a variance weight of 1 [36]. For this experiment each pixel is approximated as an independent random variable, as computing inter-pixel dependence is currently computationally intractable. The variance of the joint probability of the entire image is therefore the addition of the variance of each pixel with unitary weight. The entropy of the entire image is likewise the sum of the pixel-wise entropies as entropy is additive for independent random variables [151]. The viewpoint with the highest image variance or entropy is then selected as the next-best-view.

The two baselines used in this experiment are the random baseline, which uniformly samples over the unseen viewpoints, and FisherRF [91], which approximates the render entropy as a function of model parameter uncertainty.

The Synthetic NeRF and TUM datasets are used for this experiment. For the Synthetic NeRF dataset, the model is initially trained from 4 views. A new image is then added every 500 iterations, with an additional 100 training iterations for each added image, until a total of 10 images are selected over 10000 training iterations. This training schedule is identical to the one presented in [91].

For the TUM dataset, the model is initially trained on 150 images as the included scenes are more complex than the object-centric Synthetic NeRF dataset. A new image is then added every 100 iterations until a total of 300 images are selected over 20000 training iterations. The results for both NBV experiments are presented in Table 4.3.

From these results we see that using the proposed color variance for NBV selection outperforms the baselines on both datasets. For the Synthetic NeRF dataset, the intricate geometric detail translates to high entropy using the method presented in [91], resulting in a diverse set of NBV selections. In comparison, the TUM dataset contains regions of homo-



Figure 4.4: The rendered variance struggles to capture un-converged regions of the splat or regions of high-frequency texture, resulting in blurry variance renders (shelf) or regions of low variance (hanging plant). Once the splat has converged, the variance likewise converges to a better approximation of the true rendering error and shows sharper, more defined regions.

geneity such as desks and walls which translate into low entropy using the method presented in [91]. The viewpoints selected by this method are therefore less informative for the TUM dataset as they prioritize close-up views of small objects with intricate geometry within the room which have high entropy.

Notably, the random baseline performs surprisingly well across all experiments, with exceptional performance in the TUM dataset.

Our proposed variance rendering method estimates uncertainty of the model output directly, choosing viewpoints with high variance or entropy directly without estimating model parameter uncertainty. We propose that the reason the color variance is a better selection criteria than depth variance for the NBV task is that the evaluation criteria, PSNR, SSIM, and LPIPS, are representative of the quality of the color render. Interestingly, the rendered entropy performs worse as a next-best-view selection criterion than the variance renders, an unexpected outcome given that the model parameter entropy from [91] performs well in these experiments. This may be due to the fact that the entropy is a function of the spatial parameters of the Gaussians in the 3DGS model and does not depend on the color, depth, or semantic parameters. We highlight that our approach does not require additional overhead, and variance/entropy images are rendered at $\geq$ 2000Hz, whereas the entropy estimates from [91] are rendered at 70Hz.

4.4 Conclusion

We present a method for rendering pixel-wise uncertainty and entropy by interpreting the rendering equation as the probabilistic process. This enables the computation of higher-order moments and entropy for each pixel in the rendered image with minimal computational overhead. Render speeds remain high at $\geq$ 200Hz in comparison to existing uncertainty approximation schemes for radiance field models.

We demonstrate that our proposed approach has a stronger correlation with rendering error than existing uncertainty quantification methods. This indicates that the rendered variance approximates regions of the scene for which the 3DGS model is a poor fit. Furthermore, we demonstrate that the rendered variance is a better selection criteria than existing NBV approaches.

As the pixel-wise variance and entropy are functions of the radiance field model itself, the rendered uncertainty is prone to erroneous or blurry estimates when the model has not converged. Likewise, the pixel-wise independence approximation used in the NBV experiment does not account for the fact that a Gaussian may effect multiple pixels. Computing the true joint probability, and resultant uncertainty, over the entire image must therefore

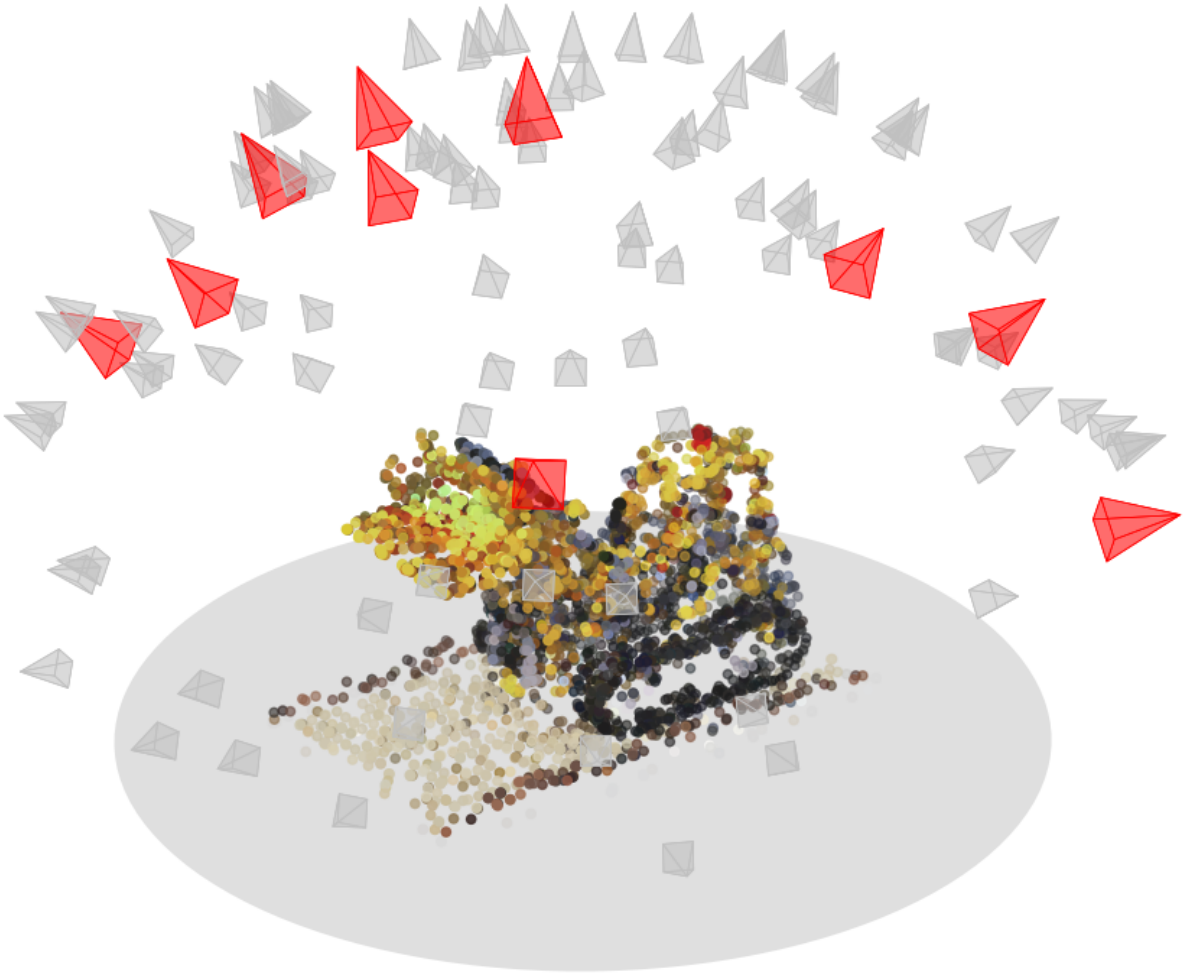


Figure 4.5: Next-Best-View for the Lego model in the Synthetic NeRF dataset. All available training cameras are shown in gray while the training cameras chosen using our proposed variance rendering approach are shown in red.

consider this dependence between pixels which is an area of future work.

CHAPTER 5

An Invariant-Preserving Sparse Spectral Discretization of the Continuum Equation

Riemannian manifolds appear in many perceptual applications for robotics. In state estimation, Lie groups provide a natural representation for quaternions which define a robot's orientation [31]. Toroidal structures have been found in the neural activation patterns of mice relating to spatial reasoning [64]. In computer vision, hyperbolic surfaces represent a natural embedding space for hierarchical semantics, where semantic similarity graphs may be projected onto these surfaces while maintaining edge distances [130]. These Riemannian manifolds also extend to other areas in robotics, where they have been used to define low-dimensional system dynamics for robotic mechanisms with differential flatness [197]. While more complex to model than Euclidean domains, Riemannian manifolds exhibit properties which are beneficial to these aforementioned tasks and provide a framework for maintaining consistency in these perceptual algorithms.

Uncertainty over these Riemannian domains is quantified and propagated in a similar manner to uncertainty in Euclidean domains. The invariant Kalman Filter [7] extends the Kalman Filter to Lie Groups and provides a means of exact state estimation under linear vector fields and Gaussian state uncertainty, and variants of this algorithm analogous to the extended or unscented Kalman Filters also exist. Unfortunately, many vector fields defined over these manifolds are nonlinear. This chapter therefore discusses a means of propagating uncertainty defined over these manifolds in the presence of nonlinear vector fields and provides a means of quantifying the accuracy of the propagated uncertainty.

5.1 Introduction

Uncertainty propagation involves advecting an initial probability density along the flow of a dynamical system via the continuum equation. Ideally, a probability advection scheme

preserves the L^1-norm, ensures positivity, and exhibits spectral convergence with sub-exponential complexity with respect to system dimension.

For linear systems with Gaussian uncertainty, uncertainty propagation is done analytically, thereby ensuring the aforementioned desired properties. For nonlinear systems, one can perform uncertainty propagation by using particles, semi-Lagrangian methods [170], particle-in-cell methods [157], or classical Monte Carlo approaches to estimate the first n-moments of the propagated density [67]. Particle methods guarantee the satisfaction of discrete analogues of measure positivity and can be designed to preserve the L^1-norm. However, particle-based approaches often suffer from slower convergence rates than their functional approximation counterparts when the uncertainty and vector field are smooth [17, 71]. Furthermore, such methods offer no meaningful error bounds on the propagated density estimates [181, 29] and scale poorly to high-dimensional systems [149].

Another alternative for uncertainty propagation involves low order approximations of the dynamical system or Gaussian approximations of uncertainty [153]. These simplifications ensure positivity, scale well with system dimension, and preserve the L^1-norm, but can result in diverging solutions, and omitting higher order terms can lead to significant errors [96].

In contrast, functional approximation methods estimate the propagated uncertainty directly. Wasserstein gradient flow methods provide a natural framework for modeling dynamics in spaces of probability measures [24], but incur a high computational cost due to the complexity of calculating the Wasserstein distance [59] and have difficulty handling high-dimensional distributions [125].

Spectral schemes converge faster than particle methods when the initial condition has some degree of regularity and the vector field is smooth [34] and may reduce the computational burden induced by functional discretization [196]. Discontinuous Galerkin methods [34] allow for flexible handling of complex geometries [140], local adaptivity [78], and high-order accuracy in solving partial differential equations [93].

Spectral methods are well-established in the study of continuum mechanics and transport equations, often viewed as special cases of first-order hyperbolic partial differential equations (PDEs) [17, 71]. Various Galerkin discretizations of the Koopman operator have been applied to generic dynamical systems [21, 119], including fluid dynamics [156], where these methods serve as a generalization of dynamic mode decomposition [159]. Similarly, Ulam-type discretizations of the Frobenius-Perron operator [107, 187] have been utilized to find invariant manifolds in systems with uniform Gaussian noise [56, 57]. In continuous time, Petrov-Galerkin discretization of the Frobenius-Perron operator's infinitesimal generator has shown convergence in the presence of noise [16] and positivity preservation, albeit at

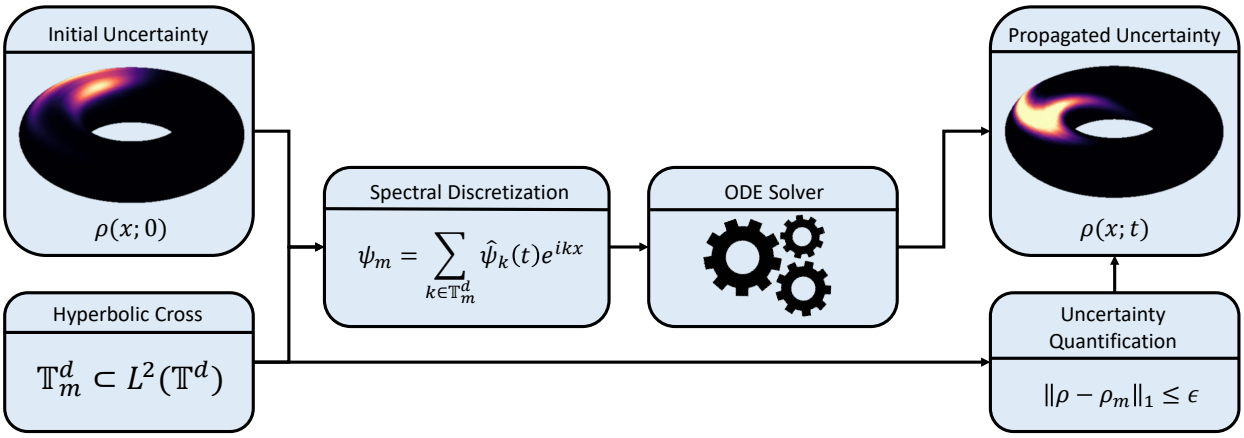


Figure 5.1: This paper formulates uncertainty propagation of a density as the advection of a sparse spectral discretization of its square-root through a semi-discretized vector field, yielding a linear ordinary differential equation. The proposed approach is then able to leverage off-the-shelf ODE solvers to compute solutions for the approximate propagated uncertainty. Furthermore, this paper proves error bounds on the accuracy of these approximations as well as on the probability measure of the propagated uncertainty.

the expense of approximation accuracy [101]. While spectral methods provide quantifiable accuracy in uncertainty propagation, they often do not guarantee positivity, L^1 preservation[1], nor sub-exponential complexity.

This chapter introduces a computational framework for propagating uncertainty for dynamical systems evolving over the d-dimensional torus. By constructing sparse spectral discretizations for functions whose squared magnitudes are densities, positivity in the propagated uncertainty is guaranteed while enabling scalability to higher dimensions via the sparse construction. The proposed method addresses the limitations of traditional spectral discretization approaches by providing quantitative bounds on approximation error, supported by simulated examples that demonstrate the framework's efficacy. Furthermore, this chapter introduces an open-source Python package, *sparselib* [43], which leverages sparsity and parallel computation for efficient, accurate, and scalable uncertainty propagation.

The remainder of the chapter is organized as follows: Section 5.2 formulates the problem of probability advection via the linear advection equation. Section 5.3 introduces the proposed sparse spectral framework for uncertainty propagation while Section 5.5 discusses the implementation details of the proposed framework. Section 5.4 presents error bounds on the approximation accuracy of the proposed approach. Section 5.6 describes the evaluation of the proposed method in simulated experiments and Section 5.7 provides concluding remarks.

[1] An example of such non-conservative behaviour is the Gibbs phenomenon.

5.1.1 Notation

Throughout the chapter, $\mathbb{T}^d$ denotes a d-dimensional torus. The Lebesgue space, L^p, is implicitly defined via the functional norm

$$\|f\|_p := \left(\int |f(x)|^p dx \right)^{1/p} \tag{5.1}$$

and the Sobolev spaces, $W^{s,p}$, via the Laplace operator [82] via the norm

$$\|f\|_{s,p} := \left(\int |(1+\Delta)^{s/2}[f](x)|^p(x) dx \right)^{1/p}. \tag{5.2}$$

Lastly, $H^s := W^{s,2}$.

5.2 Problem Formulation

Let $(\mathbb{T}^d, \mathcal{F}, P)$ denote a probability space, where $\mathcal{F}$ is the σ-algebra and P the probability measure such that

$$P(\Omega \in \mathcal{F}) = \int_\Omega \rho(x;t) \mathrm{d}x \tag{5.3}$$

where $\rho : \mathbb{T}^d \times T \to \mathbb{R}$ is a time-varying probability density function. Let X be a vector field on $\mathbb{T}^d$ given in local coordinates by $X(x) = (X^1(x), \ldots, X^d(x))$. The probability density function is transformed under the evolution of X via the continuum equation, given in local coordinates by

$$\partial_t \rho + \pounds_X[\rho] = 0 \tag{5.4}$$

where $\pounds_X[\rho] := \sum_{i=1}^d \partial_i(\rho X^i)$ denotes the Lie derivative.

Equation (5.4), which is sometimes called the "advection equation" or "Liouville's equation", describes how a density (e.g., a probability distribution) is transported by the flow of X. Such PDEs arise in a variety of contexts, ranging from mechanics [9, 183] to control theory [80], and can be seen as a zero-noise limit of the forward Kolmogorov equation [134].

The solution to (5.4) takes the form

$$\rho(x;t) = |\det(D(\Phi_X^t)^{-1}(x))| \rho((\Phi_X^t)^{-1}(x);0) := (\Phi_X^t)_* \rho(x;0) \tag{5.5}$$

where $\Phi_X^t : \mathbb{T}^d \to \mathbb{T}^d$ is the time-t flow map associated to X. One can deduce that the L^1-norm of ρ is conserved in time [108, Theorem 16.42]. The proof is an application of the

change of variables formula. Let Ω be a measurable set on $\mathbb{T}^d$, then

$$\begin{aligned}\int_\Omega \rho(x;t)\mathrm{d}x &= \int_\Omega |\det(D(\Phi_X^t)^{-1}(x))|\rho((\Phi_X^t)^{-1}(x);0)\mathrm{d}x \\ &= \int_{(\Phi_X^t)^{-1}(\Omega)} \rho(x;0)\mathrm{d}x\end{aligned} \tag{5.6}$$

This is consistent with the conservation of push-forward measure [50]. From this, it is important to note that the solutions of (5.4) have the following properties:

$$\rho(x;0) > 0 \implies \rho(\Phi_X^t(x);t) > 0 \tag{5.7}$$

$$\frac{d}{dt}\int_{\mathbb{T}^d} \rho(x;t)\mathrm{d}x = 0 \tag{5.8}$$

The property (5.7) is referred to as positivity, while the property (5.8) is referred to as mass conservation.

This chapter derives a numerical scheme for (5.4) that achieves spectral convergence with sub-exponential complexity with respect to system dimension, and exact (up to machine precision) satisfaction of positivity and mass conservation.

5.3 Uncertainty Propagation for Sparse Half Densities

This section presents a framework for solving (5.4) via a sparse spectral method that considers the evolution of a related PDE on $\mathbb{T}^d$. The solution ρ is assumed to be periodic over $\mathbb{T}^d$, enabling the representation of the probability density as a Fourier series, $\rho(x;t) = \sum_{k\in\mathbb{N}^d} \hat{\rho}_k(t)e^{ikx}$, where $\hat{\rho}_k$ are the time-varying Fourier coefficients. Substituting the Fourier discretization into (5.4), one obtains a set of coupled ordinary differential equations (ODE) for the Fourier coefficients.

When using a truncated Fourier expansion of this form, neither positivity (5.7) nor mass conservation (5.8) are guaranteed. This is not the approach we will present. Rather, in the method we shall present shortly, positivity and mass conservation will be enforced via the transformation of ρ into a *half-density*, while sub-exponential complexity with respect to dimension will be satisfied via the use of a sparse, hyperbolic cross approximation in place of a full-tensor product of Fourier modes.

5.3.1 Unitary Transformations of L^2

In this section we will consider the advection of an L^2 function, ψ, via the auxiliary linear advection equation

$$\partial_t \psi + \pounds_X^{1/2}[\psi] = 0 \tag{5.9}$$

where $\pounds_X^{1/2}[\psi] = \sum_{i=1}^d \frac{1}{2} X_i \partial_i \psi + \frac{1}{2}\partial_i(\psi X_i)$ is the Lie derivative of the half-densities.

Notably, solutions to this auxiliary PDE are useful as their squared form produces solutions to (5.4), which guarantee positivity by the following proposition.

Proposition 6. *If ψ is a solution of (5.9) then $\rho := |\psi|^2 = \psi^* \psi$ is a solution of (5.4).*

Proof. In local coordinates, the product rule is applied to find

$$\partial_i(X^i \rho) = \partial_i(X^i \psi^* \psi) = \psi \partial_i(X^i \psi^*) + \psi X^i \partial_i \psi^* \tag{5.10}$$

By a different grouping and application of the product rule, it is also shown that

$$\partial_i(X^i \rho) = \psi^* \partial_i(X^i \psi) + \psi^* X^i \partial_i \psi \tag{5.11}$$

A linear combination of these equations yields

$$\partial_i(X^i \rho) = \frac{1}{2}\psi^* \partial_i(X^i \psi) + \frac{1}{2}\psi \partial_i(X^i \psi^*) + \frac{1}{2}\psi^* X^i \partial_i \psi + \frac{1}{2}\psi X^i \partial_i \psi^* \tag{5.12}$$

This implies $\pounds_X[\rho] = \pounds_X^{1/2}[\psi]\psi^* + \pounds_X^{1/2}[\psi^*]\psi$. Then

$$\begin{aligned} \partial_t \rho = \partial_t(\psi \psi^*) &= \psi^* \partial_t \psi + \psi \partial_t \psi^* \\ &= -\pounds_X^{1/2}[\psi]\psi^* - \psi \pounds_X^{1/2}[\psi^*] = -\pounds_X[\rho] \end{aligned}$$

□

The half-density continuum equation also describes a unitary evolution on L^2 which is easy to discretize spectrally. This is in contrast to spectral discretizations of (5.4) which may not be L^1 preserving. The unitary transformation over L^2 is proven in the following proposition and it is noted in subsequent sections how this relates to L^1 preservation.

Proposition 7. *The evolution of L^2 under (5.9) is unitary.*

Proof. Let $\langle \psi, \varphi \rangle$ denote the L^2 inner product. Let $\varphi, \psi \in L^2$ be supported on a local

coordinate chart. Then

$$\begin{aligned}\langle \pounds_X^{1/2}[\psi], \overline{\varphi}\rangle &= \sum_{i=1}^{d} \int \left(\frac{1}{2} X_i \partial_i \psi + \frac{1}{2}\partial_i(\psi X_i)\right) \varphi \mathrm{d}x \\ &= -\sum_{i=1}^{d} \int \psi \left(\frac{1}{2} X_i \partial_i \overline{\varphi} + \frac{1}{2}\partial_i(\overline{\varphi} X_i)\right) \mathrm{d}x\end{aligned} \tag{5.13}$$

where the last line is obtained by invoking integration by parts. This illustrates that $\pounds_X^{1/2}$ is an anti-self-adjoint operator on L^2. The proof is expanded to the entirety of $\mathbb{T}^d$ via a partition of unity. By Stone's theorem, this implies that the evolution induced by $\pounds_X^{1/2}$ is unitary. □

5.3.1.1 A Geometric Perspective

In this subsection we provide a more principled reasoning for the use of the half-density continuum equation to describe uncertainty propagation. Readers may skip this explanation without disrupting the flow of the chapter.

All advection PDEs consider the continuous transformation of geometric objects (e.g. functions, densities, vector-fields, tensors) with respect to time. For uncertainty propagation, the geometric objects of interest are termed α-densities which are parameterized by $\alpha \in [0, 1]$. These α-densities are functions on the frame-bundle of a manifold, such that on each frame the function scales according to x^α. For a more thorough definition of α-densities, see [74, Chapter 4] [10, Appendix A].

Two values of α yield familiar objects. 0-densities are functions, for which the natural norm is the sup-norm. What analysts call densities are 1-densities for which the natural norm is the 1-norm. It is well understood that the space of densities is embedded in L^1 and that the space of functions is embedded in L^∞. It can be shown that these embeddings generalize such that the space of α-densities maps into $L^{1/\alpha}$, however this mapping is not unique.

Half-densities are embedded within L^2 spaces, and the multiplication of two half-densities within this space results in a density in L^1. This leads to the simplistic description of half-densities as "square-roots" of densities.

Finally, geometric objects are defined by how they transform under a change of coordinates, and α-densities are no exception. When α-densities are transformed under the infinitesimal motion of a vector field, the continuum equation is likewise transformed into

$$\partial_t \gamma + (1-\alpha) X_i \partial_i \gamma + \alpha \partial_i (X_i \gamma) = 0. \tag{5.14}$$

Note that (5.4) is the case where $\alpha = 1$ and (5.9) is the case where $\alpha = 1/2$.

5.3.2 Hyperbolic Cross Approximation

To compute numerical solutions to (5.9) that scale at sub-exponential rates with respect to system dimension, we introduce a discretization scheme. The basic ingredient of this discretization is a Hilbert basis that is denoted by $\{e_0, e_1, \dots\}$ for $L^2(\mathbb{T}^d)$.

To begin, the space of continuous, complex-valued functions $C(\mathbb{T}^d)$ is discretized. Classical spectral schemes construct a compact d-dimensional approximation space of $\mathbb{T}^d$ based on the isotropic full tensor product grid [142]. Let V_∞^d denote this approximation space.

For traditional isotropic full tensor product discretization schemes, the number of basis functions grows exponentially with respect to system dimension, $\mathcal{O}(N^d)$, where N is the number of basis functions in each dimension. This is the so-called *curse of dimensionality*, which is further compounded by the fact that the operator $\pounds_X^{1/2}$ grows at a rate of $\mathcal{O}(N^{2d})$ due to the construction of the matrix operator from the Galerkin equations [52]. This exponential growth quickly becomes intractable even for low-dimensional systems.

Sparse grids were originally introduced by [167] to alleviate the curse of dimensionality present in tensor product grids used for multi-dimensional quadrature and were later extended for function spaces [73, 68]. Given a half-density, ψ, these methods compute an optimal subset of V_∞^d via the restricted optimization problem

$$\min_{\substack{V_m^d \subset V_\infty^d \\ |V_m^d| = N}} \|\psi - I_m(\psi)\|_2 \tag{5.15}$$

where I_m is the projection operator of the Hilbert basis onto the sparse function space V_m^d with chosen cardinality N, and m is the so-called *maximum grid level*. The relation between N and m is

$$N = 2^m \left(\frac{m^{d-1}}{(d-1)!} + \mathcal{O}(m^{d-2}) \right). \tag{5.16}$$

From inspection, sparse tensor schemes grow at a sub-exponential rate of $\mathcal{O}\big(N(logN)^{d-1}\big)$, a significant improvement over the full tensor counterpart with only minimal degradation in terms of the approximation error [22]. Rather than discretize the domain $\mathbb{T}^d$ directly, a sparse discretization of frequency space is considered such that the corresponding Fourier bases provide optimal approximation properties. The isotropic tensor product discretization of the truncated frequency domain is denoted as $\mathbb{T}_\infty^d$.

Akin to computing an optimal, sparse approximation space subset via (5.15), the hyperbolic cross provides the optimal restriction of the frequency domain [179], where the

Algorithm 6: A spectral discretization to solve (5.9) for half densities.

Input: $\psi(x;0) \in L^2(\mathbb{T}^d)$, $t \in \mathbb{R}$, $m \in \mathbb{N}$.

initialize $\mathbb{T}^d_m$

initialize $\hat{\psi}(0) \in \mathbb{C}^N$

initialize $X_m \in \mathbb{C}^{N \times N}$

for $a = 1, \ldots, N$ **do**

  $[\hat{\psi}(0)]_a = \int_{\mathbb{T}^d} \psi(x;0) \overline{e}^{ik_a x} \mathrm{d}x$

  for $b = 1, \ldots, N$ **do**

    $[X_m]^a_b = \int_{\mathbb{T}^d} \overline{e}^{ik_a x} (X^\alpha \partial_\alpha e^{ik_b x} + \frac{1}{2} \partial_\alpha (X^\alpha \overline{e}^{ik_b x}))(x) \mathrm{d}x$

initialize the function $F : \mathbb{C}^N \to \mathbb{C}^N$ given by $F(z) = X_m \cdot z$.

$z(t) = \mathrm{OdeSolve}(F, z(0), t)$

Output: $\psi_m(x;t) = \sum_{j=1}^N [\hat{\psi}(t)]_j e^{ik_j x}$.

hyperbolic cross, $\mathbb{T}^d_m$, is defined as

$$\mathcal{J} := \left\{ j \in \mathbb{N}_0^d : \prod_{i=1}^d \max\{|j_i|, 1\} \leq 2^{m-1} \right\}, \tag{5.17}$$

$$\mathbb{T}^d_m = \mathrm{span}\,\{f_j : j \in \mathcal{J}\} \tag{5.18}$$

and π_m is the associated interpolation operator of the Hilbert basis onto $\mathbb{T}^d_m$. Computing the projection of $\pi_m(\psi)$ can be accomplished either via least squares or a hyperbolic cross fast Fourier transform [41].

5.3.3 Sparse Half-Density Density Propagation

In summary, (5.4) is transformed into (5.9) by considering half-density functions. The advection of half-densities is then transformed into a coupled set of ODEs of Fourier coefficients defined over the hyperbolic cross.

Let $\mathrm{OdeSolve}(F, x_0, t)$ denote the numerically computed solution $x(t)$ to the ODE $\dot{x} = F(x)$ at time $t \in \mathbb{R}$, with initial condition $x_0 \in \mathbb{T}^d$. Before constructing an algorithm to spectrally discretize (5.4) in a qualitatively accurate manner, (5.9) is first solved using the proposed sparse spectral discretization in Algorithm 6 [17, 146].

Algorithm 6 produces a function $\psi_m(x;t) \in \mathbb{T}^d_m$ by projecting (5.9) to $\mathbb{T}^d_m$. This projection is done by constructing the operator $X_m = \pi_m \circ \pounds_X^{1/2}|_{\mathbb{T}^d_m}$. In Section 5.4 it is shown that ψ_m converges to the solution of (5.9) as $m \to \infty$. Note that ψ_m evolves by unitary transformations, just as the exact solution to (5.9) does. This correspondence is key in providing the accuracy of the algorithms, so it is formally stated it here.

Algorithm 7: A spectral discretization to solve (5.4) for densities

Data: $\rho(x;0) \in L^1(\mathbb{T}^d), t \in \mathbb{R}, m \in \mathbb{N}$.

1 Initialize $\psi(x;0) = \sqrt{\rho^+(x;0)} - i\sqrt{\rho^-(x;0)}$;

2 Set $\psi_m(x;t) = \text{Algorithm_6}(\psi(x;0), t, m)$;

Output: $\rho_m(x;t) = \psi_m(x;t)^2$.

Proposition 8. *The output of Algorithm 6 is given by $U_m(t) \cdot \psi_m(\cdot;0)$ when $\psi(\cdot;0) \in L^2(\mathbb{T}^d)$ is the input to Algorithm 6, where $\psi_m(\cdot;0) = \pi_m(\psi(\cdot;0))$ and $U_m(t)$ is the unitary operator generated by X_m.*

Proof. The operator X_m in Algorithm 6 is anti-Hermitian on $\mathbb{T}^d_m$. It therefore generates a unitary action on $\mathbb{T}^d_m \subset L^2(\mathbb{T}^d)$ when inserted into OdeSolve. □

Given the propagated half-density output of Algorithm 6, Algorithm 7 is invoked to compute the approximate propagated density, ρ_m. In Section 5.4 it is proven that ρ_m converges to the solution of (5.4) as $m \to \infty$.

5.4 Error Analysis on the Torus

This section derives error bounds for Algorithms 6 and 7. Recall that k^2 is the k-th eigenvalue of the Laplacian on the circle (i.e., the 1-torus). From this, the definition of the mix-norm is specified [72] for $\mathbb{T}^d$ in the standard Fourier basis. The mix-norm of the Hilbert space $\mathcal{H}^{s',\ell}_{mix} \subset \mathcal{H}^\ell$ is given as

$$\|\psi\|^2_{mix,s',\ell} := \sum_{k \in \mathbb{N}_0^d} \mathcal{F}[\psi](k) \left(\prod_{i=1}^d (1 + k_i^2) \right)^{2s'} \left(1 + \sum_{i=1}^d k_i^2 \right)^{2\ell} \tag{5.19}$$

where $\psi \in L^2$ and $\mathcal{F}[\psi]$ is the Fourier transform of ψ.

This norm is then leveraged to upper-bound the interpolation error for the initial half-density.

Proposition 9 (Theorem 5 of [72])**.** *Let $\psi \in \mathcal{H}^{s',\ell}_{mix}$. Then the interpolation error is*

$$\|\psi - \pi_m(\psi)\|_s \leq C(d) N^{-(\ell - s + s')/d} \|\psi\|_{mix,s',\ell} \tag{5.20}$$

where π_m is the interpolant of ψ with respect to the the H^s norm and N is the cardinality of the approximation space.

Notably, the growth of $C(d)$ may decay for certain values of s' and ℓ [72].

A bound for the propagated half-density is then computed for Algorithm 6.

Theorem 10. *Let ψ_t be the solution of (5.9) with initial condition $\psi_0 \in \mathcal{H}^{s',\ell}_{mix}$. Let $\tilde{\psi}_t := \pi_m(\psi_t)$ be the output of Algorithm 6. Then, for $t > 0$*

$$\|\psi_t - \tilde{\psi}_t\|_s \leq C(d)N^{-(\ell-s+s')/d}\left(\|\psi_0\|_{mix,s',\ell} + t\|\pounds_X^{1/2}[\psi_0]\|_{mix,s',\ell}\right) \tag{5.21}$$

Proof. Let $\varepsilon(t) := \|\psi_t - \tilde{\psi}_t\|_s$. Let $\langle\cdot,\cdot\rangle_s$ denote the inner-product on H^s. Then

$$\frac{d\varepsilon}{dt} = \frac{1}{\varepsilon}\,\mathrm{Re}\langle\partial_t\psi - \partial_t\tilde{\psi},\, \psi - \tilde{\psi}\rangle_s \tag{5.22}$$

Applying the Cauchy-Schwarz inequality this provides an upper bound

$$\frac{d\varepsilon}{dt} \leq \|\partial_t\psi - \partial_t\tilde{\psi}\|_s \tag{5.23}$$

By assumption, $\partial_t\psi = -\pounds_X^{1/2}[\psi]$. Given that $\tilde{\psi}$ is the output of Algorithm 6, then $\partial_t\tilde{\psi} = (\pi_m \circ \pounds_X^{1/2})[\tilde{\psi}]$. For brevity, denote $\tilde{\pounds_X}^{1/2} := \pi_m \circ \pounds_X^{1/2}$. The bound then becomes

$$\frac{d\varepsilon}{dt} \leq \|\pounds_X^{1/2}[\psi] - \tilde{\pounds_X}^{1/2}[\tilde{\psi}]\|_s \tag{5.24}$$

$$= \|(\pounds_X^{1/2}[\psi] - \pounds_X^{1/2}[\tilde{\psi}]) + (\pounds_X^{1/2}[\tilde{\psi}] - \tilde{\pounds_X}^{1/2}[\tilde{\psi}])\|_s. \tag{5.25}$$

This is upper bounded by the triangle inequality

$$\frac{d\varepsilon}{dt} \leq \|\pounds_X^{1/2}[\psi - \tilde{\psi}]\|_s + \|(\pi_m^{\perp} \circ \pounds_X^{1/2})[\tilde{\psi}]\|_s \tag{5.26}$$

Integrating this bound in time around $t = 0$ provides a Lipshitz inequality of

$$\varepsilon(t) \leq \|\psi_0 - \tilde{\psi}_0\|_s + Ct\left(\|\pounds_X^{1/2}[\psi_0 - \tilde{\psi}_0]\|_s + \|(\pi_m^{\perp} \circ \pounds_X^{1/2})[\tilde{\psi}_0]\|_s\right) \tag{5.27}$$

for some $C > 0$ where $\pi_m^{\perp} := 1 - \pi_m$ is the projection operator orthogonal to the interpolant operator π_m. The first term is the approximation error at time $t = 0$. Note that $(\pi_m^{\perp}\pounds_X^{1/2}) \circ [\tilde{\psi}]$ is the approximation of $\pounds_X^{1/2}[\tilde{\psi}]$ while $\|\pounds_X^{1/2}[\psi - \tilde{\psi}\|_s \leq \|\pounds_X^{1/2}\|_{\mathrm{op}}\|\psi - \tilde{\psi}\|_s$ by the use of an operator norm on $\pounds_X^{1/2}$ as a map from H^{s+1} to H^s. Proposition 9 is invoked to derive the desired result by applying it to ψ_0 and $\pounds_X^{1/2}[\psi_0]$. □

From Theorem 10, it is seen that the approximation error decreases as $N \to \infty$, and from (5.16), this implies $m \to \infty$.

As Algorithm 6 is related to Algorithm 7 by the act of squaring the half-density output,

the error bound of Theorem 10 is converted into a bound for Algorithm 7.

Theorem 11. *Let ρ_t be a solution of (5.4) and $\tilde{\rho}_t := |\tilde{\psi}_t|^2$ be the corresponding output of Algorithm 7. Then, for $t > 0$*

$$\|\rho_t - \tilde{\rho}_t\|_1 \leq \|\psi_t - \tilde{\psi}_t\|_s$$

Proof. The first part of Algorithm 2 is the image of Algorithm 6, related by the fact that a $|\psi|^2$ is a non-negative member of L^1 whenever $\psi \in L^2$. Let ψ_t be a solution of (5.9) such that $|\psi_t|^2 = \rho_t$. Let $\tilde{\psi}_t$ the output of algorithm 6 with respect to ψ_t so that $\tilde{\rho} := |\tilde{\psi}|^2$ is the output of algorithm 7. Then

$$\begin{aligned} \|\rho - \tilde{\rho}\|_1 &= \||\psi|^2 - |\tilde{\psi}|^2\|_1 \\ &= \|(\psi - \tilde{\psi})(\psi + \tilde{\psi})\|_1. \end{aligned}$$

By Holder's inequality

$$\|\rho - \tilde{\rho}\| \leq \|\psi - \tilde{\psi}\|_2 \cdot \|\psi + \tilde{\psi}\|_2$$

This is a product of two terms. The first is bounded by Theorem 10 and the second is shown to be bounded by a multiple of $\|\rho\|_1^{1/2}$. Because $\|\rho_t\|_1$ is constant in time and given that $\|\rho_0\|_1 = 1$ by the definition of probability measure, the result follows. □

In combination with Theorem 10, Theorem 11 shows the accuracy of the discretization scheme improves as $m \to \infty$.

Finally, the bound on Algorithm 7 is used to place a error bound on the pushforward measure approximation.

Theorem 12. *Let ρ_t be a solution of (5.4) and $\tilde{\rho}_t := |\tilde{\psi}_t|^2$ be the corresponding output of Algorithm 7. Let $\Omega \subset \mathbb{T}^d$ be a measurable set. Then the pushforward measure approximation is bounded by*

$$\int_\Omega \rho_t \leq \int_\Omega \tilde{\rho}_t + \|\rho_t - \tilde{\rho}_t\|_1 \tag{5.28}$$

Proof. Expanding the left-hand side and applying Theorem 11 yields the bound

$$\begin{aligned} \int_\Omega \rho_t &= \int_\Omega \rho_t + \tilde{\rho}_t - \tilde{\rho}_t \\ &\leq \int_\Omega \tilde{\rho}_t + \|\rho_t - \tilde{\rho}_t\|_1 \end{aligned} \tag{5.29}$$

□

Table 5.1: Number of basis functions for the hyperbolic cross, $\mathbb{T}^2_m$, and the full product tensor grid, $\mathbb{T}^2_\infty$ with varying grid level m.

	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=9$
$\lvert\mathbb{T}^2_m\rvert$	5	17	49	137	349	865	2069	4841
$\lvert\mathbb{T}^2_\infty\rvert$	9	25	81	289	1089	4225	16641	66049

Table 5.2: Number of basis functions for the hyperbolic cross, $\mathbb{T}^3_m$, and the full product tensor grid, $\mathbb{T}^3_\infty$ with varying grid level m.

	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$
$\lvert\mathbb{T}^3_m\rvert$	31	111	375	1163	3391	9387
$\lvert\mathbb{T}^3_\infty\rvert$	125	729	4913	35937	274625	2146689

5.5 Implementation Details

Algorithms 6 and 7 are implemented within the open-source Python package *sparselib* [43]. This package can be installed either via the *pip* package installer or manually following the instructions on the project webpage[2]. Scripts for all experiments in Section 5.6 are also provided.

sparselib is written in a modular manner, making it possible to change the basis functions used in the discretization, enable an adaptive hyperbolic cross approximation, or other problem-specific adaptations. Furthermore, all functions in *sparselib* are written via constructive algorithms with the express purpose of avoiding the curse of dimensionality from generate-and-test algorithms. Tables 5.1-5.3 denote the number of Fourier basis functions for varying state dimension and maximum grid level for the hyperbolic cross approximation space alongside a standard full tensor product approximation space.

The computation of the projection operator, π_m, follows a least squares formulation. Evaluation points for this orthogonal projection correspond to the sparse grid points from V^d_m which are computed via the maximum grid level. For reference on how to construct this sparse grid, see [22]. Sparse least squares can be viewed as a sparse quadrature scheme, and the approximation accuracy is bounded via [22, Theorem 3.8], satisfying Proposition 9.

Table 5.3: Number of basis functions for the hyperbolic cross, $\mathbb{T}^6_m$, and the full product tensor grid, $\mathbb{T}^6_\infty$ with varying grid level m.

	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$
$\lvert\mathbb{T}^6_m\rvert$	97	545	2681	12165	51233	199869
$\lvert\mathbb{T}^6_\infty\rvert$	15625	531441	2.41e7	1.29e9	7.5e10	4.61e12

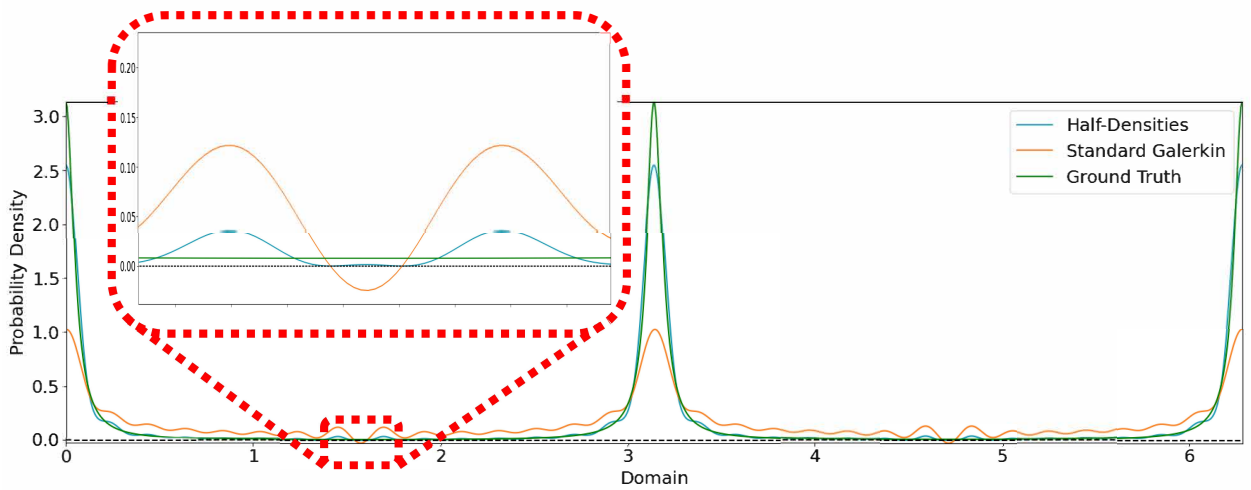


Figure 5.2: Uncertainty propagation through a 1-dimensional non-linear system with known ground-truth posterior and an initial uniform uncertainty, $\rho(x;0) \sim \mathcal{U}[0, 2\pi]$. The sparse spectral approach using our auxiliary pde maintains high accuracy while errors in the standard Galerkin approach cascade as the uncertainty is propagated.

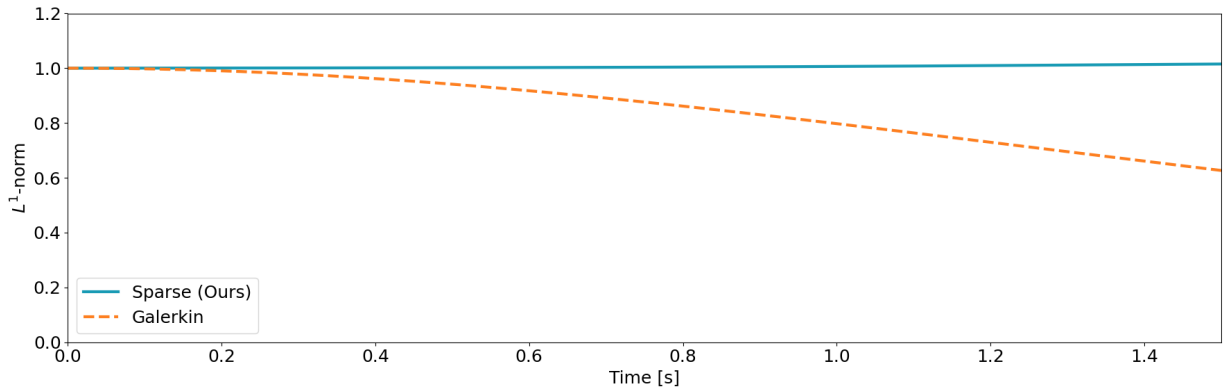


Figure 5.3: The L^1-norm of the spectral discretization for the proposed sparse and standard spectral methods for the benchmark T^1 flow. The proposed approach is L^1-norm preserving whereas the standard spectral approach is not.

5.6 Numerical Experiments

This section demonstrates the proposed sparse spectral discretization scheme on several non-linear dynamical systems with varying dimension. First, a non-linear system evolving over $\mathbb{T}^1$ with known pushforward density is introduced and the proposed method is compared to a standard Galerkin discretization scheme.

Next, the proposed approach is compared against a baseline Monte Carlo approach and a standard Galerkin discretization scheme for a modified ABC flow over $\mathbb{T}^2$ and $\mathbb{T}^3$. It is shown that the proposed sparse spectral scheme has improved qualitative accuracy compared to the standard spectral approach with the same maximum grid level. Finally, the proposed approach is compared to a Monte Carlo method for a non-chaotic harmonic oscillator based on the Kuramoto model defined over $\mathbb{T}^6$. It is demonstrated that the proposed approach has favorable scaling across varying maximum grid levels and this scaling is contrasted against a full tensor approach. Note, all experiments are conducted on a laptop with a 5.4GHz i9-13900HX processor and 32GB of RAM.

5.6.1 Benchmark $\mathbb{T}^1$ Flow

This experiment demonstrates the quantitative accuracy, and positivity and mass conservation properties of the proposed half-density formulation.

Consider the vector field $\dot{x} = -\sin(2x)$ for $x \in \mathbb{T}^1$. The flow field of this system is given by

$$\Phi_X^t(x) = \arctan(e^{2t}\tan(x)). \tag{5.30}$$

Given an initial uniform density, $\rho(x;0) = dx$, the exact solution to (5.4) is:

$$\rho(x;t) = (e^{2t}\sin^2(x) + e^{-2t}\cos^2(x))^{-1}|dx|. \tag{5.31}$$

Figure 5.2 depicts the evolution of the propagated uncertainty after 1.5s given the initial uniform condition. The exact solution, the standard spectral approach, and the proposed sparse spectral approach, both using a maximum grid level of $m = 7$, are shown. Note that in 1-dimension both the full tensor and sparse grid schemes have the same number of Fourier basis functions. Differences in the performance of these methods for this experiment are therefore due to the use of the unitary evolution of half-densities induced by (5.9).

Figure 5.2 shows that the proposed sparse spectral approach has greater quantitative accuracy than a standard spectral discretization. Notably, the standard Galerkin method

[2] https://github.com/ParkerEwen5441/sparselib

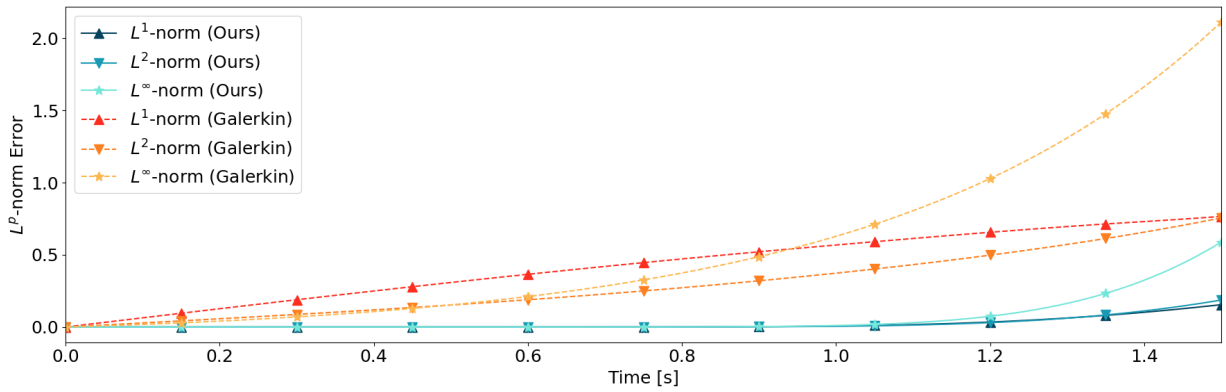


Figure 5.4: The L^p-norm errors of the standard Galerkin approach and the proposed sparse spectral method for the 1-dimensional $\mathbb{T}^1$ flow with known ground-truth. A maximum grid level of $m = 6$ is used. The proposed sparse spectral approach maintains a lower L^p-norm error compared to the standard Galerkin method for the entire experiment duration, demonstrating the utility of the auxiliary equation (5.9) in uncertainty propagation. Note, in this 1-dimensional example the hyperbolic cross is equivalent to the standard Fourier discretization.

exhibits negative probability density, which is not achievable by the exact system, whereas the proposed scheme ensures positivity of probability measure. Furthermore, Figure 5.4 shows that the L^p norm of the error signal is significantly smaller for the proposed sparse scheme, indicating better accuracy for probability computation of the pushforward density via Theorem 11.

Lastly, the spectral convergence of Algorithm 7 is shown in Figure 5.5. With respect to numerical accuracy, the proposed approach has a lower coefficient of convergence than the standard Galerkin approach. This is once again due to the non-positivity of the propagated uncertainty, whereas the proposed approach guarantees positivity via the use of the auxiliary (5.9).

5.6.2 Modified ABC flow over $\mathbb{T}^2$ and $\mathbb{T}^3$ Domains

These experiments demonstrate the qualitative accuracy of the sparse, hyperbolic approximation space and the positivity-preserving property of the half-density formulation.

Consider the system

$$\begin{aligned} \dot{x}_1 &= A\sin(x_2) + B\cos(x_1) \\ \dot{x}_2 &= C\sin(x_2) + A\cos(x_1) \end{aligned} \tag{5.32}$$

on the 2-torus such that $x, y \in \mathbb{T}^2$ for constants $A, B, C \in \mathbb{R}$. For the following numerical

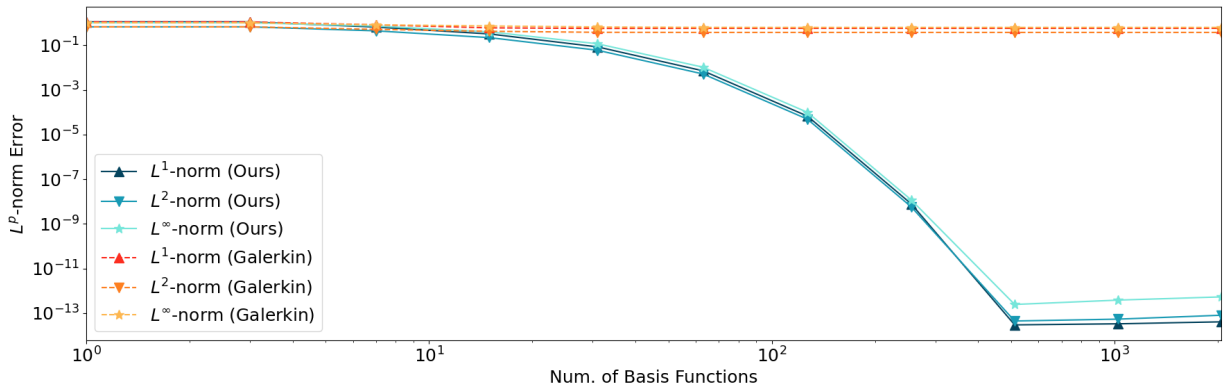


Figure 5.5: The convergence of the L^p-norm errors of the propagated uncertainty at time $t = 1s$ for the 1-dimensional benchmark $\mathbb{T}^1$ flow with known ground-truth (5.31). The proposed sparse spectral approach maintains a lower L^p-norm error compared to the standard spectral method even when the same number of basis functions are used. Notably, increasing the number of Fourier basis functions does not improve the accuracy of the propagated uncertainty in the standard spectral scheme.

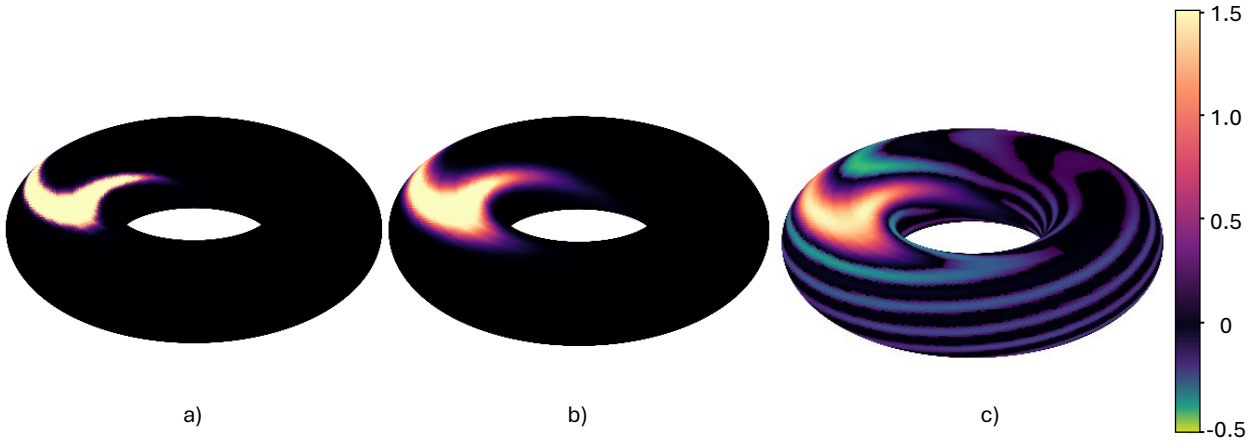


Figure 5.6: Uncertainty propagation for 1.5s through a modified ABC flow (5.32) over $\mathbb{T}^2$. a) The Monte Carlo baseline is used to compare the b) proposed sparse spectral approach and the c) standard spectral scheme. The proposed sparse spectral approach is qualitatively more accurate than the standard spectral approach even with 4.9**x** fewer basis functions. Note that the standard Galerkin approach has regions of negative probability density while the proposed sparse scheme does not.

experiment let $A = 1.0$, $B = 0.2$, and $C = 0.5$. As an initial condition consider a wrapped Normal distribution centered at $\mu = (\pi, \pi)$ with anisotropic variance $\sigma = (0.2, 0.3)$. Figure 5.6 shows approximate solutions of (5.4) using a Monte Carlo baseline with 5000 particles,

a standard spectral method, and the proposed sparse spectral approach from Algorithm 7. In this example, a maximum grid level of $m = 7$ is used for both the standard spectral method and the proposed sparse spectral method. Kernel density estimation (KDE) is used to approximate the probability density from the propagated Monte Carlo samples. Note that the KDE method accurately approximates the first moment of the propagated distribution, however the smoothing factor in KDE is a tunable hyperparameter and can result in varying higher-order moments. From Figure 5.6, it is shown that the proposed approach is qualitatively similarly to the Monte Carlo method with respect to the first moment. Furthermore, there are regions of negative probability density for the standard spectral approach which are not achievable in the true system. Table 5.1 denotes the number of basis functions for the sparse and full tensor product grids for this 2-dimensional experiment with varying grid level m.

Next, consider the system

$$\begin{aligned} \dot{x}_1 &= A\sin(x_3) + C\cos(x_2) + D\cos(x_1) \\ \dot{x}_2 &= B\sin(x_3) + A\cos(x_2) + D\cos(x_2) \\ \dot{x}_3 &= A\sin(x_3) + B\cos(x_2) + D\cos(x_3) \end{aligned} \tag{5.33}$$

on the 3-torus for constants $A, B, C, D \in \mathbb{R}$. When $D = 0$, this system is the well-studied volume conserving system known as the Arnold-Beltrami-Childress flow. When $A > B > C > 0$, $D = 0$, and $C << 1$, the solutions of this ODE are chaotic, resulting in a uniform steady state distribution. When $D = 0$, the operator $\pounds_X^{1/2}$ is identical to the operator $\pounds_X$ that appears in (5.4) and Algorithm 6 does not differ from a standard spectral discretization. Therefore, the case where $D > 0$ is considered to compare how the proposed sparse discretization of (5.9) differs from (5.4). When $D > 0$, volume is no longer conserved and there is a non-uniform steady-state distribution.

For the following numerical experiment, let $A = 1.0$, $B = 0.5$, $C = 0.2$, and $D = 0.5$. A Normal distribution centered at $\mu = (\pi, \pi, \pi)$ with an anisotropic variance of $\sigma = (0.2, 0.3, 0.3)$ is used as the initial condition and this is propagated for 1.5 seconds. Figure 5.7 depicts the x_3-marginal of the propagated densities over $\mathbb{T}^2$ for the Monte Carlo baseline using 10000 particles and the proposed sparse spectral discretization scheme with a maximum grid level of $m = 7$. Note that even for this low-dimensional problem the standard spectral method is infeasible at $m = 7$ as it requires 2.4Tb of memory. In contrast, the sparse spectral method only requires 365Mb of memory. Due to the computation limitations of the standard spectral approach, we are unable to compare against it.

Table 5.2 denotes the number of basis functions for the sparse and full tensor product grids

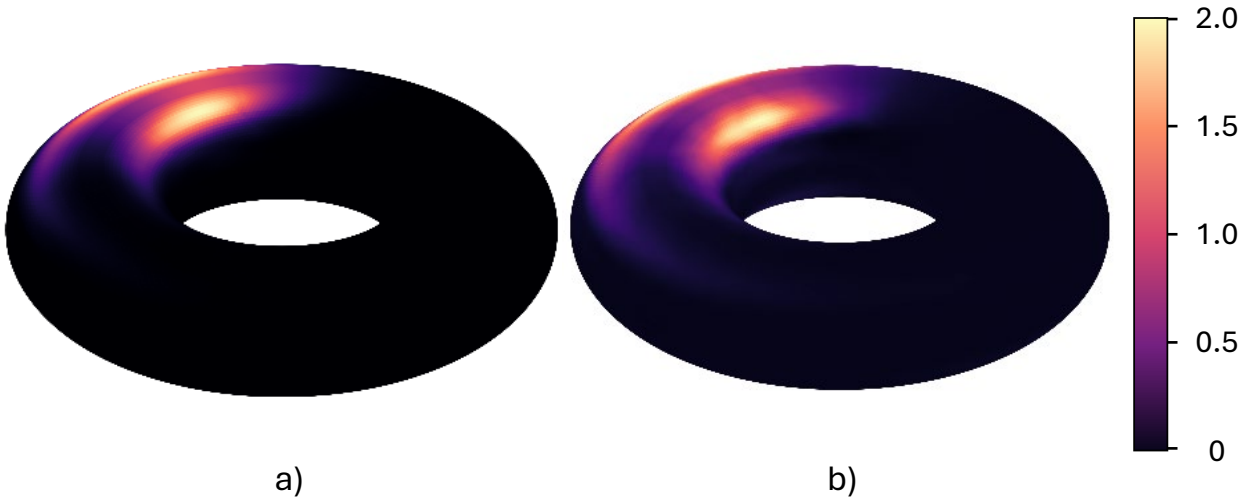


Figure 5.7: Uncertainty propagation for $t = 1.5$s through the ABC flow (5.33) over $\mathbb{T}^3$ marginalized onto $\mathbb{T}^2$ for visualization. a) The baseline Monte Carlo approach is compared against the b) proposed sparse spectral method with a max level $m = 7$. Note, the sparse spectral discretization scheme uses **81x** fewer basis functions than a standard Galerkin scheme at this grid level.

for this 3-dimensional experiment with varying grid level m. Even for this low dimensional system, the sparse grid has an order-of-magnitude fewer basis functions than the full tensor approach for $m > 4$.

Lastly, we compare the computation times of the proposed sparse spectral approach and the standard spectral approach in Figure 5.8. Both methods are implemented using the *sparselib* [43] Python package. As is expected, the proposed sparse spectral approach requires less time to propagate uncertainty as the hyperbolic cross approximation uses fewer basis functions than the standard spectral scheme.

5.6.3 Harmonic Oscillator in $\mathbb{T}^6$

This experiment demonstrates the sub-exponential scaling property of the sparse, hyperbolic approximation space.

Consider the system

$$\dot{x}_i = \omega_i + \sum_{j=1}^{N} K_{ij} \sin(x_j - x_i), \quad i = 1, \ldots, 6 \tag{5.34}$$

on the 6-torus. This system is the well-studied Kuramoto model [1] composed of 6 coupled

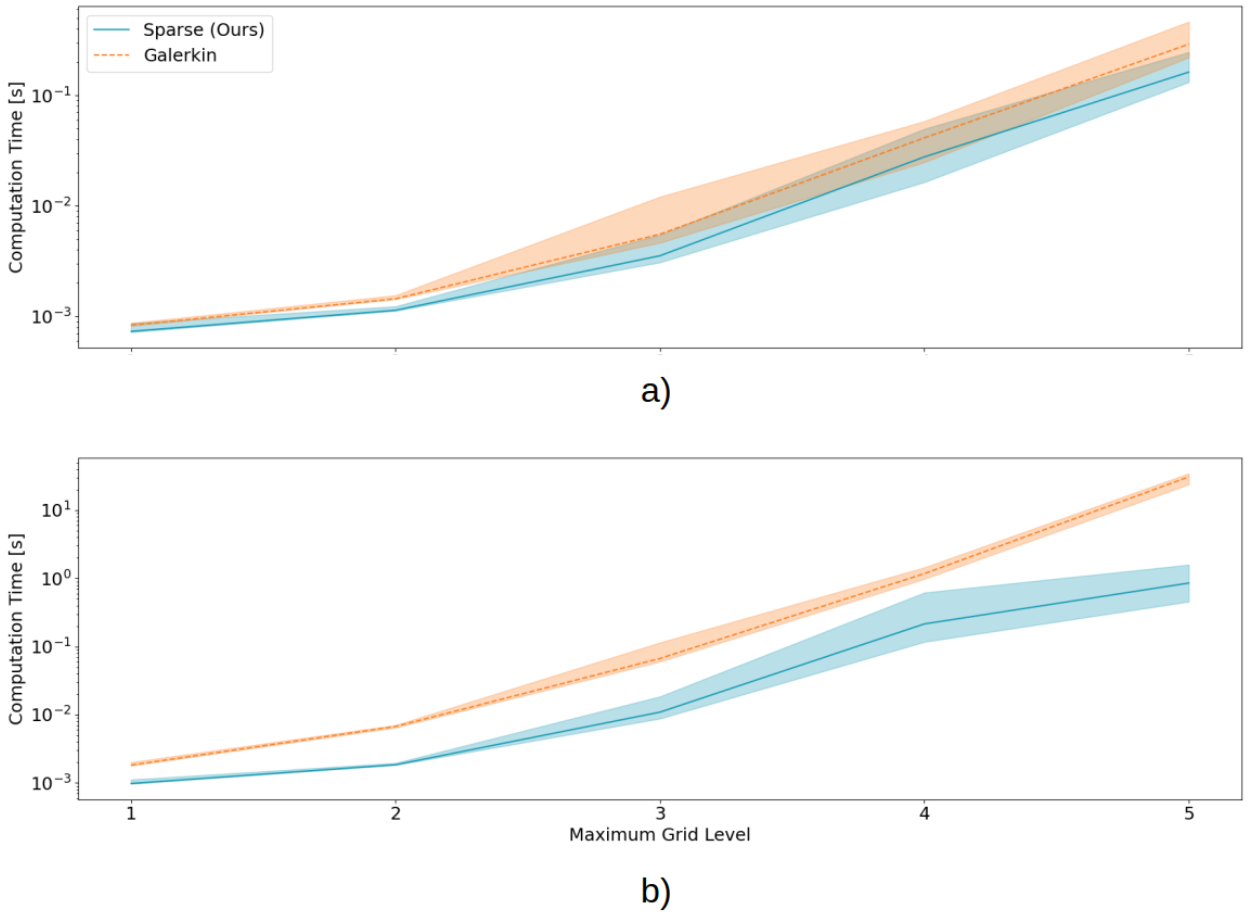


Figure 5.8: A comparison of the average computation times and the associated three standard deviations for the a) 2D and b) 3D experiments for the proposed sparse and standard spectral methods computed using 100 trials. For the 3-dimensional system the hyperbolic approximation space results in an order-of-magnitude reduction in computation time for larger grid levels.

harmonic oscillators with natural frequencies ω_i and an off-diagonal, symmetric matrix K. To ensure non-chaotic behaviour, small coupling constants are chosen such that all non-zero elements of K are equal to 0.1 and the natural frequencies are set to $\omega_i = \pi$. A Normal distribution centered at $\mu = (\pi, \pi, \pi, \pi, \pi, \pi)$ with an isotropic variance of $\sigma = (0.3, 0.3, 0.3, 0.3, 0.3, 0.3)$ is used as the initial condition and this is propagated for 0.5 seconds.

Table 5.3 denotes the number of basis functions for the sparse and full tensor product grids for this 6-dimensional experiment with varying grid level m. This demonstrates the exponential increase in the number of basis functions needed for a full tensor product approach. For this example, the standard spectral approach requires 1.1Tb of memory for a grid level of $m = 4$. In comparison, the proposed approach requires only 1.2Mb of memory

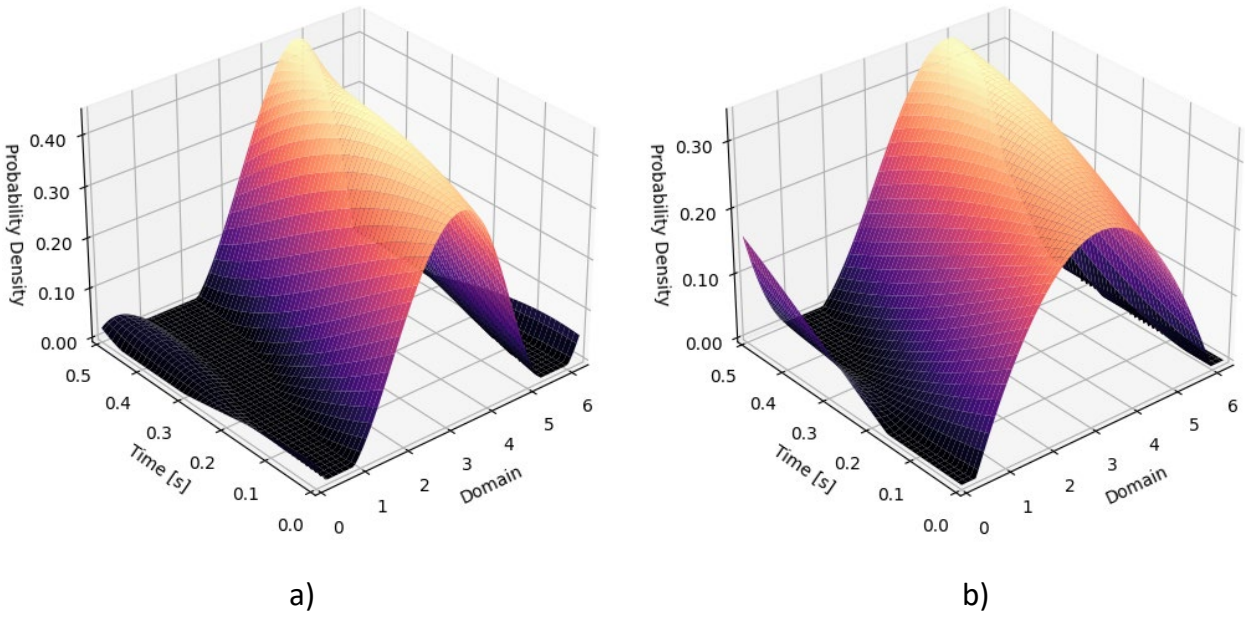


Figure 5.9: Uncertainty propagation for $t = 0.5$s for the Kuramoto model (5.34) over $\mathbb{T}^6$. A grid level of $m = 4$ is used. The marginalized, time-varying uncertainty of the first mode of the system is shown for a) the Monte Carlo approach and b) the proposed sparse grid approach. Even in 6-dimensional space the sparse spectral discretization scheme is able to run with limited computational resources with results comparable to that of Monte Carlo.

for an equivalent grid level.

The marginalized, time-varying first mode of the system is shown in Figure 5.9 alongside the Monte Carlo approach using $100,000$ particles. A sparse quadrature approach was used to marginalize the sparse spectral system. Note the proposed sparse spectral scheme has comparable results to the Monte Carlo approach and is able to run with limited computational resources. More accurate results are obtained when the grid level is increased, however this also increases the computational complexity required to marginalize the propagated uncertainty.

5.7 Conclusion

This chapter proposes a sparse spectral discretization method for uncertainty propagation through nonlinear vector fields over the d-dimensional torus via half-densities. The standard continuum equation for probability densities is first transformed into the continuum equation for the corresponding half-densities. A hyperbolic cross approximation is then used to discretize the space of half-density functions whose domain is the d-dimensional torus.

Solutions to the sparse, half-density continuum equation are then computed via Algorithm 6 and the corresponding propagated densities are computed via Algorithm 7. The proposed method provides convergence proofs and upper bounds the propagated uncertainty error. It is shown that the proposed method outperforms the standard spectral discretization scheme across several numerical examples and has comparable performance to a Monte Carlo simulation baseline in cases where the standard spectral approach cannot be computed due to computational memory limitations.

CHAPTER 6

Conclusion

In robotics, understanding and managing uncertainty is essential to achieving reliable and robust performance across a wide range of tasks. This dissertation has explored uncertainty in geometric and semantic contexts that are critical for robotic perception, decision-making, and interaction in complex environments. By addressing the inherent unpredictability in sensor data and environmental variability, these models quantify uncertainty and enable robots to operate more effectively in real-world scenarios.

Geometric uncertainty for surface mesh and radiance field models provides a foundation for robots to safely navigate and manipulate objects. For contact surface meshes, triangle vertex heights are predicted via Bayesian inference. For radiance field models, pixels of the rendered image are treated as independent random variables, for which the structure of the rendering equation enables the computation of the moments and entropy of each pixel.

Semantic uncertainty complements geometric uncertainty quantification. By modeling semantic and property uncertainty, robotic systems can manage the ambiguity that arises from complex visual inputs, varying lighting conditions, occlusions, and stochastic machine learning model outputs. This dissertation has demonstrated that incorporating probabilistic methods into semantic segmentation and classification tasks not only reduces semantic prediction error but also improves the robot's adaptability and resilience. Multiple sensing modalities, such as touch and vision, can also be represented as observations, and Bayesian inference the enables closed-form, computationally tractable semantic and property uncertainty quantification.

Lastly, this work has extended the propagation of uncertainty into higher-dimensional spaces, specifically over the d-torus, allowing for uncertainty propagation in non-Euclidean spaces. Half-densities and a sparse functional discretization were considered to construct invariant-preserving spectral framework for propagating uncertainty. This framework was compared against existing state-of-the-art approaches and we demonstrated improvements for L^1 norm preservation, positivity preservation, computation times, and accuracy. Quan-

titative error bounds were also derived, enabling a priori computation of the quality of the spectral framework.

In conclusion, this dissertation demonstrates the value of a comprehensive approach to uncertainty modeling in robotics. By addressing both geometric and semantic uncertainties, and extending these concepts to Riemannian manifolds, it offers a robust framework for enhancing the safety, flexibility, and intelligence of robotic systems. As future robotic applications continue to push the boundaries of autonomy and interaction, uncertainty-aware models will be indispensable for enabling robots to operate seamlessly in dynamic and complex environments. The research presented herein contributes to this ongoing effort, building a foundation for more resilient and contextually aware robotic systems.

6.1 Future Directions

6.1.1 Low-Dimensional Data Representations

Finding efficient, low-dimensional data representations is crucial for robotics, as these representations enhance the speed, robustness, and reliability of perception and planning algorithms in complex environments. Using low-dimensional feature spaces in learned models improves generalization, interpretability, and adaptability, enabling systems to perform better with less data and computational demand [130, 11].

Riemannian manifolds serve as a natural domain in robotics for various task-relevant data, such as semantic similarity [130], waypoint-based mapping structures in the brain [64], and configuration space transformations [31]. The curved geometry of these manifolds require specialized methods for effective data representation. For example, in transforming a robot's configuration on a Lie Group, Gaussian priors enable closed-form solutions for linear vector fields [6], but nonlinear or non-Gaussian systems require numerical methods. Unfortunately, conventional numerical techniques over Riemannian manifolds can be computationally intensive and may fail to preserve essential invariants, such as positivity or the L^1 norm [180].

This dissertation presents an invariant-preserving numerical scheme for uncertainty propagation over Riemannian manifolds. This framework leverages sparsity and functional analysis to achieve efficient spectral convergence and lower computational costs, with derived bounds on accuracy and convergence rates. Compared to current state-of-the-art methods, my approach improves accuracy and significantly reduces computational time. This work is available as open-source software[1].

[1] https://roahmlab.github.io/sparselib_page/

The presented invariant-preserving numerical scheme for uncertainty propagation over Riemannian manifolds has applications for these aforementioned low-dimensional, Riemannian domains. Machine learning models for vision and robotics often ignore Riemannian priors, instead relying on large quantities of data to learn approximations of these natural domain embeddings. However, such high-dimensional approximations are inefficient for robotics, where systems must react in real-time to external stimuli. Future work in this area will first focus on developing low-dimensional Riemannian feature spaces for task-relevant data representations in semantic mapping and navigation, leveraging the hyper-ball, Poincaré disk, and torus to capture underlying structure while preserving uncertainty. This approach seeks to enhance the learning efficiency of neural networks and facilitate robust decision-making in resource-constrained robotic systems. Next, uncertainty over these Riemannian domains will be quantified, enabling the representation of uncertainty of learned model outputs and sensor noise. The propagation of this uncertainty may then be achieved via the work presented in Chapter 5.

6.1.2 Active Perception Via Reinforcement Learning

While machine learning has achieved remarkable progress in the past decade, robotics has not experienced comparable leaps forward. The reason for this, and a source of numerous challenges in robotics, is the inherent uncertainty in perceptual data. For a robot to reason about its surroundings it relies on this noisy sensor data. Thus, the algorithms employed for perceptual understanding should account for this uncertainty rather than ignore it and further enable a robot to reason about this uncertainty.

Mapping from images or point clouds enables mobile robots to perform object avoidance and terrain traversal, and embedding object, or semantic, labels within these maps provides further information about a robot's surroundings. Semantic labels are often computed using learning-based computer vision methods, but these methods are non-recursive, meaning they cannot use priors to refine estimates, and, as such, these methods are not robust while dealing with noisy sensor data. The work presented in Chapters 2 and 3 provides a means of quantifying the uncertainty of semantic label predictions [198, 201] and leveraging this uncertainty for robust and safe task completion [121].

While semantic mapping has remained a dominant form of robot perception, it only offers one type of information for planning algorithms. As demonstrated in this dissertation, it is the physical properties of a robot's environment which are the driving factors for robotic task success such as legged locomotion [44] or manipulation [45, 121]. To address the problem of estimating these properties a method for quantifying uncertainty in tactile and visual

sensing modalities using probabilistic inference was presented, enabling robots to quantify uncertainty in these sensor readings for more accurate semantic and property predictions in real time [44, 45]. This property estimation framework enables robots to be more robust to terrain changes, such as walking onto ice.

To bridge the gap between semantic mapping and the challenges of real-world legged locomotion, future work will need to integrate uncertainty quantification in tactile and visual sensing with reinforcement learning, enabling robots to account for perceptual noise and make safer, more robust decisions in complex environments. In a world built for human locomotion, legged robots have an advantage over traditional wheeled or aerial platforms. Unfortunately, the drawback of legged platforms is that they are inherently unstable, an issue which is further complicated by the complex interactions at the robot's feet as it traverses its environment. Reinforcement learning has recently shown incredible performance for legged locomotion, however existing methods do not account for perceptual noise in the robot's sensors. While these reinforcement learning techniques have demonstrated excellent performance in simulation, they struggle to leverage existing computer vision techniques on hardware as these methods fail to quantify uncertainty in their visual predictions, leading to overconfident or dangerous robot behaviors.

Future work in this direction will build of the presented work in Chapters 2 and 3 in uncertainty quantification for property estimation [47] and legged locomotion planning and control with a mobile manipulator [48] to enable reinforcement learning agents to leverage the uncertainty within semantic map representations to accomplish robust legged locomotion. Furthermore, leveraging the correct inductive biases of this semantic and property data will further assist in providing a structured environment for such reinforcement learning algorithms, enabling these models to generalize more effectively with less data. By combining uncertainty quantification, reinforcement learning, and the correct data-relevant feature representation, legged robots will have increased locomotive abilities, be able to avoid dangerous situations, and find regions of the environment for which more information is required for task success, thus enabling long-horizon autonomy.

6.1.3 Fusion of Multiple Sensing Modalities

The primary aim of the previous future research direction was to enable robots to learn how to learn via active and interactive perception via visual data. Inherent in this is the notion of a robot interacting with its environment and perceiving the effect of these interactions. Notably, this requires multiple sensing modalities: a means of sensing and measuring interactions through touch, and a means of perceiving the effects of these interactions through

vision or another sensing modality. A robot is not limited to two sensing modalities however, and additional sensing modalities will enable a more thorough collection of task-relevant environmental data.

This dissertation presented a mean of uncertainty quantification for semantic mapping and property prediction in Chapters 2 and 3, offering a means of leveraging both a robot's visual sensors and tactile sensors. By modeling the relationship between an object's material classifications, collected from visual measurements, and its physical properties, measured through tactile sensing, the robot was able to reason about these multiple sensing modalities together. This approach enables fast, computationally efficient property estimation and can be used across multiple different robotic platforms, from autonomous off-road vehicles to legged robots. Similar to how humans use both touch and vision to learn about the world [12], this dissertation shows that tactile sensing improves the property prediction accuracy when compared to methods which only use vision [44, 47]. The proposed algorithm is evaluated across multiple robotic platforms, including off-road autonomous vehicles [198], legged robots [47], and manipulators [44, 45] and demonstrates the broad utility of this approach. The developed algorithm is also shown to enable legged robots to traverse hazardous terrain more reliably when compared to other state-of-the-art robotic mapping methods [44] and aiding in opening doors, further motivating the utility of this approach for long-horizon autonomy.

While it was shown that touch and vision are complimentary sensing modalities, this research direction aims to leverage additional modalities, such as thermal cameras, event cameras, and frequency-modulated continuous wave (FMCW) lidar, to enable robots to perceive additional task-relevant properties about their environments. While certain properties may be best observed using a camera, other properties, such as motion, are better observed using FMCW lidars, while still others, such as mass, are best observed via tactile force-torque sensors. By creating representations for which these sensing modalities can be reasoned about in tandem, the robot will be able to choose the optimally informative sensor to learn about its environment to maximize task success likelihood.

This future work direction builds off the previous two. The first direction aims is to find the correct low-dimensional feature representation for each sensing modality while simultaneously enabling uncertainty quantification over these low-dimensional embedding spaces. The second direction will determine how to condition these sensor measurements on one another. For example, to relate tactile and vision modalities I conditioned surface friction estimates on material classifications. Such measurement conditioning allows for a cohesive and context-aware fusion of information across multiple sensing modalities, where the strengths of each modality compliment one another. In this final research direction, the aim is to extend the theory proposed in this dissertation to reinforcement learning for legged robots to incorpo-

rate interactive perception, enabling agents to not only adapt to new environments but also to refine their own learning processes over time. By integrating active exploration, these robots will develop the ability to identify and prioritize relevant information autonomously, learning how to learn as they encounter varied, complex, and previously unseen scenarios. This approach will empower robots to become increasingly efficient in gathering task-relevant information, accelerating their adaptability and robustness in dynamic environments.

BIBLIOGRAPHY

[1] Juan A Acebrón, Luis L Bonilla, Conrad J Pérez Vicente, Félix Ritort, and Renato Spigler. The kuramoto model: A simple paradigm for synchronization phenomena. Reviews of modern physics, 77(1):137–185, 2005.

[2] Ruslan Agishev, Tomáš Petříček, and Karel Zimmermann. Trajectory optimization using learned robot-terrain interaction model in exploration of large subterranean environments. IEEE Robotics and Automation Letters, 7(2):3365–3371, 2022.

[3] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. Advances in neural information processing systems, 33:14927–14937, 2020.

[4] Brian Anderson and John B Moore. Optimal filtering. Courier Corporation, 2012.

[5] Anelia Angelova, Larry Matthies, Daniel Helmick, and Pietro Perona. Slip prediction using visual information. In Robotics: Science and Systems, 08 2006.

[6] Axel Barrau and Silvère Bonnabel. The invariant extended Kalman filter as a stable observer. IEEE Transactions on Automatic Control, 62(4):1797–1812, 2016.

[7] Axel Barrau and Silvere Bonnabel. Invariant kalman filtering. Annual Review of Control, Robotics, and Autonomous Systems, 1(1):237–257, 2018.

[8] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In Proceedings of the IEEE/CVF international conference on computer vision, pages 5855–5864, 2021.

[9] George Keith Batchelor. An introduction to fluid dynamics. Cambridge university press, 1967.

[10] Sean Bates and Alan Weinstein. Lectures on the Geometry of Quantization, volume 8. American Mathematical Soc., 1997.

[11] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261, 2018.

[12] Michael S Beauchamp, Nafi E Yasar, Richard E Frye, and Tony Ro. Touch, sound and vision in human superior temporal sulcus. Neuroimage, 41(3):1011–1020, 2008.

[13] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In Proceedings of the IEEE International Conference on Computer Vision, 2019.

[14] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In Proceedings of the IEEE International Conference on Computer Vision, 2019.

[15] José M Bernardo and Adrian FM Smith. Bayesian theory, volume 405. John Wiley & Sons, 2009.

[16] Andreas Bittracher, Péter Koltai, and Oliver Junge. Pseudogenerators of spatial transfer operators. SIAM Journal on Applied Dynamical Systems, 14(3):1478–1517, 2015.

[17] John P Boyd. Chebyshev and Fourier spectral methods. Courier Corporation, 2001.

[18] Martim Brandao, Kenji Hashimoto, and Atsuo Takanishi. Friction from vision: A study of algorithmic and human performance with consequences for robot perception and teleoperation. In 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), pages 428–435. IEEE, 2016.

[19] Martim Brandão, Kenji Hashimoto, and Atsuo Takanishi. Friction from vision: A study of algorithmic and human performance with consequences for robot perception and teleoperation. In International Conference on Humanoid Robots, pages 428–435, 2016.

[20] Zachary Brei, Jonathan Michaux, Bohao Zhang, Patrick Holmes, and Ram Vasudevan. Serving time: Real-time, safe motion planning and control for manipulation of unsecured objects. IEEE Robotics and Automation Letters, 2024.

[21] Marko Budišić, Ryan Mohr, and Igor Mezić. Applied koopmanism. Chaos: An Interdisciplinary Journal of Nonlinear Science, 22(4), 2012.

[22] Hans-Joachim Bungartz and Michael Griebel. Sparse grids. Acta numerica, 13:147–269, 2004.

[23] Xiaoyi Cai, Siddharth Ancha, Lakshay Sharma, Philip R Osteen, Bernadette Bucher, Stephen Phillips, Jiuguang Wang, Michael Everett, Nicholas Roy, and Jonathan P How. Evora: Deep evidential traversability learning for risk-aware off-road autonomy. arXiv preprint arXiv:2311.06234, 2023.

[24] Jose A Carrillo, Daniel Matthes, and Marie-Therese Wolfram. Lagrangian schemes for wasserstein gradient flows. Handbook of Numerical Analysis, 22:271–311, 2021.

[25] Vincent Cartillier, Zhile Ren, Neha Jain, Stefan Lee, Irfan Essa, and Dhruv Batra. Semantic MapNet: Building Allocentric SemanticMaps and Representations from Egocentric Views. In Proceedings of the AAAI National Conference on Artificial Intelligence, February 2021.

[26] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niebner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3D: Learning from RGB-D Data in Indoor Environments. In Proceedings of the IEEE International Conference on 3D Vision, pages 667–676, 2017.

[27] Guikun Chen and Wenguan Wang. A survey on 3D Gaussian Splatting. arXiv preprint arXiv:2401.03890, 2024.

[28] Jiaqi Chen, Zeyu Yang, and Li Zhang. Semantic segment anything. https://github.com/fudan-zvg/Semantic-Segment-Anything, 2023.

[29] Yen-Chi Chen. A tutorial on kernel density estimation and recent advances. Biostatistics & Epidemiology, 1(1):161–187, 2017.

[30] Yizhou Chen, Yiting Zhang, Zachary Brei, Tiancheng Zhang, Yuzhen Chen, Julie Wu, and Ram Vasudevan. Differentiable discrete elastic rods for real-time modeling of deformable linear objects. Conf. Robot. Learning., 2024.

[31] Claude Chevalley. Theory of Lie groups. Courier Dover Publications, 2018.

[32] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Proc. Conf. Empirical Methods in NLP, pages 1724–1734, 2014.

[33] Jaeyoung Chung, Jeongtaek Oh, and Kyoung Mu Lee. Depth-regularized optimization for 3d gaussian splatting in few-shot images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 811–820, 2024.

[34] Bernardo Cockburn and Chi-Wang Shu. The runge–kutta discontinuous galerkin method for conservation laws v: multidimensional systems. Journal of computational physics, 141(2):199–224, 1998.

[35] Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. Noise reduction in speech processing, pages 1–4, 2009.

[36] Emile Contal, David Buffoni, Alexandre Robicquet, and Nicolas Vayatis. Parallel gaussian process optimization with upper confidence bound and pure exploration. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 225–240. Springer, 2013.

[37] Camille Couprie, Clément Farabet, Laurent Najman, and Yann LeCun. Indoor semantic segmentation using depth information. arXiv preprint arXiv:1301.3572, 2013.

[38] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[39] Persi Diaconis and Donald Ylvisaker. Conjugate priors for exponential families. The Annals of statistics, pages 269–281, 1979.

[40] Kevin Doherty, Tixiao Shan, Jinkun Wang, and Brendan Englot. Learning-Aided 3-D Occupancy Mapping with Bayesian Generalized Kernel Inference. IEEE Trans. Robot., 35(4):953–966, 2019.

[41] Michael Döhler, Stefan Kunis, and Daniel Potts. Nonequispaced hyperbolic cross fast fourier transform. SIAM journal on numerical analysis, 47(6):4415–4428, 2010.

[42] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In Conference on Robot Learning, pages 1 – 16, 2017.

[43] Parker Ewen. sparselib: A python library for uncertainty propagation via sparse spectral discretizations. https://roahmlab.github.io/sparselib_page/, 2024.

[44] Parker Ewen, Hao Chen, Yuzhen Chen, Anran Li, Anup Bagali, Gitesh Gunjal, and Ram Vasudevan. You've got to feel it to believe it: Multi-modal bayesian inference for semantic and property prediction. Robotics: Science and Systems, 2024.

[45] Parker Ewen, Gitesh Gunjal, Hao Chen, Anran Li, Yuzhen Chen, and Ram Vasudevan. Multi-modal semantic perception using Bayesian inference. IEEE IROS Workshop on Integrated Perception, Planning, and Control for Physically and Contextually-Aware Robot Autonomy, 2023. EAISI Best Poster Presentation Award.

[46] Parker Ewen, Gitesh Gunjal, Joey Wilson, Jinsun Liu, Challen Enninful Adu, and Ram Vasudevan. Not all actions are created equal: Bayesian optimal experimental design for safe and optimal nonlinear system identification. arXiv preprint arXiv:2308.01829, 2023.

[47] Parker Ewen, Adam Li, Yuxin Chen, Steven Hong, and Ram Vasudevan. These maps are made for walking: Real-time terrain property estimation for mobile robots. IEEE Robotics and Automation Letters, 7(3):7083–7090, 2022.

[48] Parker Ewen, Jean-Pierre Sleiman, Yuxin Chen, Wei-Chun Lu, Marco Hutter, and Ram Vasudevan. Generating continuous motion and force plans in real-time for legged mobile manipulation. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pages 4933–4939. IEEE, 2021.

[49] Péter Fankhauser, Michael Bloesch, and Marco Hutter. Probabilistic terrain mapping for mobile robots with uncertain localization. IEEE Robotics and Automation Letters, 3(4):3019–3026, 2018.

[50] Alessio Figalli and Federico Glaudo. An invitation to optimal transport, Wasserstein distances, and gradient flows. EMS Textbooks In Mathematics, 2021.

[51] Paul Filitchkin and Katie Byl. Feature-based terrain classification for littledog. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1387–1392. IEEE, 2012.

[52] Clive AJ Fletcher and CAJ Fletcher. Computational galerkin methods. Springer, 1984.

[53] Peter I Frazier. A tutorial on Bayesian optimization. arXiv preprint arXiv:1807.02811, 2018.

[54] Jonas Frey, David Hoeller, Shehryar Khattak, and Marco Hutter. Locomotion policy guided traversability learning using volumetric representations of complex environments. In 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5722–5729. IEEE, 2022.

[55] Jonas Frey, Matias Mattamala, Nived Chebrolu, Cesar Cadena, Maurice Fallon, and Marco Hutter. Fast traversability estimation for wild visual navigation. arXiv preprint arXiv:2305.08510, 2023.

[56] Gary Froyland, Oliver Junge, and Péter Koltai. Estimating long-term behavior of flows without trajectory integration: The infinitesimal generator approach. SIAM Journal on Numerical Analysis, 51(1):223–247, 2013.

[57] Gary Froyland and Kathrin Padberg. Almost-invariant sets and invariant manifolds—connecting probabilistic and geometric descriptions of coherent structures in flows. Physica D: Nonlinear Phenomena, 238(16):1507–1523, 2009.

[58] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In international conference on machine learning, pages 1050–1059. PMLR, 2016.

[59] Thomas O Gallouët and Leonard Monsaingeon. A jko splitting scheme for kantorovich–fisher–rao gradient flows. SIAM Journal on Mathematical Analysis, 49(2):1100–1130, 2017.

[60] Lu Gan, Youngji Kim, Jessy W. Grizzle, Jeffrey M. Walls, Ayoung Kim, Ryan M. Eustice, and Maani Ghaffari. Multi-task learning for scalable and dense multi-layer bayesian map inference. ArXiv, abs/2106.14986, 2021.

[61] Lu Gan, Youngji Kim, Jessy W Grizzle, Jeffrey M Walls, Ayoung Kim, Ryan M Eustice, and Maani Ghaffari. Multitask learning for scalable and dense multilayer bayesian map inference. IEEE Transactions on Robotics, 39(1):699–717, 2022.

[62] Lu Gan, Ray Zhang, Jessy W. Grizzle, Ryan M. Eustice, and Maani Ghaffari. Bayesian Spatial Kernel Smoothing for Scalable Dense Semantic Mapping. IEEE Robot. Autom. Letter., 5(2):790–797, 2020.

[63] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, Pablo Martinez-Gonzalez, and Jose Garcia-Rodriguez. A survey on deep learning techniques for image and video semantic segmentation. Applied Soft Computing, 70:41–65, 2018.

[64] Richard J Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A Baas, Benjamin A Dunn, May-Britt Moser, and Edvard I Moser. Toroidal topology of population activity in grid cells. Nature, 602(7895):123–128, 2022.

[65] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 3354–3361, 2012.

[66] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In Proceedings of the Asian Conference on Computer Vision, pages 25–38, 2011.

[67] Andrew Gelman. Method of moments using monte carlo simulation. Journal of Computational and Graphical Statistics, 4(1):36–54, 1995.

[68] Thomas Gerstner and Michael Griebel. Numerical integration using sparse grids. Numerical algorithms, 18(3-4):209–232, 1998.

[69] James J Gibson. The theory of affordances. Hilldale, USA, 1(2), 1977.

[70] Lily Goli, Cody Reading, Silvia Sellán, Alec Jacobson, and Andrea Tagliasacchi. Bayes' Rays: Uncertainty quantification for neural radiance fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 20061–20070, 2024.

[71] David Gottlieb and Jan S Hesthaven. Spectral methods for hyperbolic problems. Journal of Computational and Applied Mathematics, 128(1-2):83–131, 2001.

[72] Michael Griebel and Jan Hamaekers. Sparse grids for the schrödinger equation. ESAIM: Mathematical Modelling and Numerical Analysis, 41(2):215–247, 2007.

[73] Michael Griebel, Micha Schneider, and Christoph Zenger. A combination technique for the solution of sparse grid problems. Forschungsberichte, TU Munich, TUM I 9038:1–24, 1990.

[74] Victor Guillemin and Shlomo Sternberg. Geometric asymptotics, volume 14. American Mathematical Soc., 1990.

[75] Nitesh B Gundavarapu, Divyansh Srivastava, Rahul Mitra, Abhishek Sharma, and Arjun Jain. Structured aleatoric uncertainty in human pose estimation. In CVPR Workshops, volume 2, page 2, 2019.

[76] Yanming Guo, Yu Liu, Theodoros Georgiou, and Michael S Lew. A review of semantic segmentation using deep neural networks. International journal of multimedia information retrieval, 7:87–93, 2018.

[77] Shijie Hao, Yuan Zhou, and Yanrong Guo. A brief survey on semantic segmentation with deep learning. Neurocomputing, 406:302–321, 2020.

[78] Ralf Hartmann and Paul Houston. Adaptive discontinuous galerkin finite element methods for nonlinear hyperbolic conservation laws. SIAM Journal on Scientific Computing, 24(3):979–1004, 2003.

[79] Paul S Heckbert. Fundamentals of texture mapping and image warping, 1989.

[80] Didier Henrion and Milan Korda. Convex computation of the region of attraction of polynomial control systems. IEEE Transactions on Automatic Control, 59(2):297–312, 2013.

[81] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural Computation, 9(8):1735–1780, 1997.

[82] Lars Hörmander. The analysis of linear partial differential operators. i. classics in mathematics, 2003.

[83] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. Autonomous Robots, 2013.

[84] Jiarui Hu, Xianhao Chen, Boyin Feng, Guanglin Li, Liangjing Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. CG-SLAM: Efficient dense RGB-D SLAM in a consistent uncertainty-aware 3d Gaussian field. arXiv preprint arXiv:2403.16095, 2024.

[85] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual Language Maps for Robot Navigation. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 10608–10615, 2023.

[86] Eddy Ilg, Özgün Çiçek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. Uncertainty Estimates and Multi-hypotheses Networks for Optical Flow. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, Proceedings of the European Conference on Computer Vision, pages 677–693, 2018.

[87] Seth Isaacson, Pou-Chun Kung, Mani Ramanagopal, Ram Vasudevan, and Katherine A Skinner. Loner: Lidar only neural representations for real-time slam. IEEE Robotics and Automation Letters, 2023.

[88] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In Proceedings of the 24th annual ACM symposium on User interface software and technology, pages 559–568, 2011.

[89] Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are Bayesian neural network posteriors really like? In International conference on machine learning, pages 4629–4640. PMLR, 2021.

[90] Priyank Jaini and Pascal Poupart. Online and distributed learning of gaussian mixture models by bayesian moment matching. arXiv preprint arXiv:1609.05881, 2016.

[91] Wen Jiang, Boshu Lei, and Kostas Daniilidis. Fisherrf: Active view selection and uncertainty quantification for radiance fields using fisher information. arXiv preprint arXiv:2311.17874, 2023.

[92] Liren Jin, Xieyuanli Chen, Julius Rückin, and Marija Popović. Neu-nbv: Next best view planning using uncertainty estimation in image-based neural rendering. In 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 11305–11312. IEEE, 2023.

[93] Martin Käser and Michael Dumbser. A highly accurate discontinuous galerkin method for complex interfaces between solids and moving fluids. Geophysics, 73(3):T23–T35, 2008.

[94] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics, 42(4):1–14, 2023.

[95] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Jeff Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d Gaussian Splatting as Markov Chain Monte Carlo. arXiv preprint arXiv:2404.09591, 2024.

[96] Masoud Khodarahmi and Vafa Maihami. A review on kalman filter models. Archives of Computational Methods in Engineering, 30(1):727–747, 2023.

[97] Jack Kiefer. General equivalence theory for optimum designs (approximate theory). The annals of Statistics, pages 849–879, 1974.

[98] Dong-Ki Kim, Daniel Maturana, Masashi Uenoyama, and Sebastian Scherer. Season-invariant semantic segmentation with a deep multimodal network. In Field and Service Robotics: Results of the 11th International Conference. Springer, 2018.

[99] Dongshin Kim, Jie Sun, Sang Min Oh, James M Rehg, and Aaron F Bobick. Traversability classification using unsupervised on-line visual learning for outdoor robot navigation. In International Conference on Robotics and Automation, pages 518–525. IEEE, 2006.

[100] Dongshin Kim, Jie Sun, Sang Min Oh, James M Rehg, and Aaron F Bobick. Traversability classification using unsupervised on-line visual learning for outdoor robot navigation. In Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., pages 518–525. IEEE, 2006.

[101] Péter Koltai. Efficient approximation methods for the global long-term behavior of dynamical systems: theory, algorithms and examples. Logos Verlag Berlin GmbH, 2011.

[102] Igor Kononenko. Bayesian neural networks. Biological Cybernetics, 61(5):361–370, 1989.

[103] Ioannis Kostavelis and Antonios Gasteratos. Semantic mapping for mobile robotics tasks: A survey. Robotics and Autonomous Systems, 66:86–103, 2015.

[104] Benjamin Kuipers and Yung-Tai Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. Robotics and Autonomous Systems, 8(1):47–63, 1991.

[105] Benjamin Kuipers and Yung-Tai Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. Robotics and autonomous systems, 8(1-2):47–63, 1991.

[106] Pou-Chun Kung, Seth Isaacson, Ram Vasudevan, and Katherine A Skinner. Sad-gs: Shape-aligned depth-supervised gaussian splatting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2842–2851, 2024.

[107] Andrzej Lasota and Michael C Mackey. Chaos, fractals, and noise: stochastic aspects of dynamics, volume 97. Springer Science & Business Media, 2013.

[108] John M Lee. Introduction to Smooth Manifolds. Springer, 2012.

[109] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven Semantic Segmentation. In Proceedings of the International Conference on Learning Representations, 2022.

[110] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pages 740–755. Springer, 2014.

[111] Jinsun Liu, Challen Enninful Adu, Lucas Lymburner, Vishrut Kaushik, Lena Trang, and Ram Vasudevan. Radius: Risk-aware, real-time, reachability-based motion planning. arXiv preprint arXiv:2302.07933, 2023.

[112] Gabriel B Margolis, Xiang Fu, Yandong Ji, and Pulkit Agrawal. Learning to see physical properties with active sensing motor policies. arXiv preprint arXiv:2311.01405, 2023.

[113] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. Journal of the American statistical Association, 46(253):68–78, 1951.

[114] Carlos Mastalli, Michele Focchi, Ioannis Havoutis, Andreea Radulescu, Sylvain Calinon, Jonas Buchli, Darwin G. Caldwell, and Claudio Semini. Trajectory and foothold optimization using low-dimensional models for rough terrain locomotion. In 2017 IEEE International Conference on Robotics and Automation, pages 1096–1103, 2017.

[115] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In 2017 IEEE International Conference on Robotics and automation (ICRA), pages 4628–4635. IEEE, 2017.

[116] Nis Meinert, Jakob Gawlikowski, and Alexander Lavin. The unreasonable effectiveness of deep evidential regression. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, pages 9134–9142, 2023.

[117] Arman Melkumyan and Fabio Ramos. A Sparse Covariance Function for Exact Gaussian Process Inference in Large Datasets. In Proceedings of the International Joint Conference on Artifical Intelligence, page 1936–1942, 2009.

[118] Arman Melkumyan and Fabio Tozeto Ramos. A sparse covariance function for exact gaussian process inference in large datasets. In Int. Joint Conf. Artif. Intell., 2009.

[119] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. Nonlinear Dynamics, 41:309–325, 2005.

[120] Jonathan Michaux, Patrick Holmes, Bohao Zhang, Che Chen, Baiyue Wang, Shrey Sahgal, Tiancheng Zhang, Sidhartha Dey, Shreyas Kousik, and Ram Vasudevan. Can't touch this: Real-time, safe motion planning and control for manipulators under uncertainty. arXiv preprint arXiv:2301.13308, 2023.

[121] Jonathan Michaux, Seth Isaacson, Challen Enninful Adu, Adam Li, Rahul Kashyap Swayampakula, Parker Ewen, Sean Rice, Katherine A Skinner, and Ram Vasudevan. Let's make a splan: Risk-aware trajectory optimization in a normalized gaussian splat. arXiv preprint arXiv:2409.16915, 2024.

[122] Jonathan Michaux, Adam Li, Qingyi Chen, Che Chen, Bohao Zhang, and Ram Vasudevan. Safe planning for articulated robots using reachability-based obstacle avoidance with spheres. Robotics: Science and Systems, 2024.

[123] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM, 65(1):99–106, 2021.

[124] Thomas Minka. Estimating a dirichlet distribution, 2000.

[125] Petr Mokrov, Alexander Korotin, Lingxiao Li, Aude Genevay, Justin M Solomon, and Evgeny Burnaev. Large-scale wasserstein gradient flows. Advances in Neural Information Processing Systems, 34:15243–15256, 2021.

[126] Vikram Mullachery, Aniruddh Khera, and Amir Husain. Bayesian neural networks. arXiv preprint arXiv:1801.07710, 2018.

[127] Raul Mur-Artal, J. Montiel, and Juan Tardos. Orb-slam: a versatile and accurate monocular slam system. IEEE Trans. Robot., 31:1147 – 1163, 10 2015.

[128] Michael Neunert, Markus Stäuble, Markus Giftthaler, Carmine D. Bellicoso, Jan Carius, Christian Gehring, Marco Hutter, and Jonas Buchli. Whole-body nonlinear model predictive control through contacts for quadrupeds. IEEE Robotics and Automation Letters, 3(3):1458–1465, 2018.

[129] Tran Nguyen, Francesco Verdoja, Fares Abu-Dakka, and Ville Kyrki. Probabilistic surface friction estimation based on visual and haptic measurements. IEEE Robotics and Automation Letters, PP:1–1, 02 2021.

[130] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. Advances in neural information processing systems, 30, 2017.

[131] Mark S. Nixon and Alberto S. Aguado. Feature extraction & image processing for computer vision (third edition). In Mark S. Nixon and Alberto S. Aguado, editors, Feature Extraction & Image Processing for Computer Vision (Third Edition), pages 489–518. Academic Press, Oxford, 2012.

[132] Donghun Noh, Hyunwoo Nam, Min Sung Ahn, Hosik Chae, Sangjoon Lee, Kyle Gillespie, and Dennis Hong. Surface material dataset for robotics applications (smdra): A dataset with friction coefficient and rgb-d for surface segmentation. In International Conference on Pattern Recognition, pages 6275–6281, 2021.

[133] Donghun Noh, Hyunwoo Nam, Min Sung Ahn, Hosik Chae, Sangjoon Lee, Kyle Gillespie, and Dennis Hong. Surface material dataset for robotics applications (smdra): A dataset with friction coefficient and rgb-d for surface segmentation. In 2020 25th International Conference on Pattern Recognition (ICPR), pages 6275–6281. IEEE, 2021.

[134] Bernt Oksendal. Stochastic differential equations: an introduction with applications. Springer Science & Business Media, 2013.

[135] Helen Oleynikova, Alexander Millane, Zachary Taylor, Enric Galceran, Juan Nieto, and Roland Siegwart. Signed distance fields: A natural representation for both mapping and planning. In RSS Workshop: Geometry and Beyond-Representations, Physics, and Scene Understanding for Robotics. University of Michigan, 2016.

[136] Utku Ozbulak, Hyun Jung Lee, Beril Boga, Esla Timothy Anzaku, Homin Park, Arnout Van Messem, Wesley De Neve, and Joris Vankerschaver. Know your self-supervised learning: A survey on image-based generative and discriminative training. arXiv preprint arXiv:2305.13689, 2023.

[137] Ghazaleh Panahandeh, Erik Ek, and Nasser Mohammadiha. Road friction estimation for connected vehicles using supervised machine learning. In Intelligent Vehicles Symposium, pages 1262–1267. IEEE, 2017.

[138] Panagiotis Papadakis. Terrain traversability analysis methods for unmanned ground vehicles: A survey. Engineering Applications of Artificial Intelligence, 26(4):1373–1385, 2013.

[139] Panagiotis Papadakis. Terrain traversability analysis methods for unmanned ground vehicles: A survey. Engineering Applications of Artificial Intelligence, 26(4):1373–1385, 2013.

[140] Per-Olof Persson and Jaime Peraire. Sub-cell shock capturing for discontinuous galerkin methods. In 44th AIAA aerospace sciences meeting and exhibit, page 112, 2006.

[141] Vladimir S Petrović and Costas S Xydeas. Sensor noise effects on signal-level image fusion performance. Information fusion, 4(3):167–183, 2003.

[142] Albrecht Pietsch. Approximation spaces. Journal of Approximation Theory, 32(2):115–134, 1981.

[143] Julio A Placed and José A Castellanos. A General Relationship between Optimality Criteria and Connectivity Indices for Active Graph-SLAM. IEEE Robot. Autom. Letter., 8(2):816–823, 2022.

[144] Julio A. Placed, Jared Strader, Henry Carrillo, Nikolay Atanasov, Vadim Indelman, Luca Carlone, and José A. Castellanos. A Survey on Active Simultaneous Localization and Mapping: State of the Art and New Frontiers. IEEE Trans. Robot., 39:1686–1705, 2022.

[145] Rudra P K Poudel, Stephan Liwicki, and Roberto Cipolla. Fast-scnn: Fast semantic segmentation network, 2019.

[146] William H Press, William T Vetterling, Saul A Teukolsky, and Brian P Flannery. Numerical recipes. Citeseer, 1988.

[147] Michael Procopio, Jane Mulligan, and Greg Grudic. Learning terrain segmentation with classifier ensembles for autonomous robot navigation in unstructured environments. Journal of Field Robotics, 26:145 – 175, 02 2009.

[148] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 20051–20060, 2024.

[149] Paul Bui Quang, Christian Musso, and Francois Le Gland. An insight into the issue of dimensionality in particle filtering. In 2010 13th International Conference on Information Fusion, pages 1–8. IEEE, 2010.

[150] AKM Rabby and Chengcui Zhang. Beyondpixels: A comprehensive review of the evolution of neural radiance fields. arXiv preprint arXiv:2306.03000, 2023.

[151] Alfréd Rényi. On measures of entropy and information. In Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, volume 1: contributions to the theory of statistics, volume 4, pages 547–562. University of California Press, 1961.

[152] Safoura Rezapour Lakani, Antonio J Rodríguez-Sánchez, and Justus Piater. Towards affordance detection for robot manipulation using affordance for parts and parts for affordance. Autonomous Robots, 43:1155–1172, 2019.

[153] Maria Isabel Ribeiro. Kalman and extended kalman filters: Concept, derivation and properties. Institute for Systems and Robotics, 43(46):3736–3741, 2004.

[154] Jeffrey S Rosenthal. First look at rigorous probability theory, A. World Scientific Publishing Company, 2006.

[155] Stergios I Roumeliotis, Gaurav S Sukhatme, and George A Bekey. Sensor fault detection and identification in a mobile robot. In Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No. 98CH36190), volume 3, pages 1383–1388. IEEE, 1998.

[156] Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S Henningson. Spectral analysis of nonlinear flows. Journal of fluid mechanics, 641:115–127, 2009.

[157] Henri Samuel. A deformable particle-in-cell method for advective transport in geodynamic modelling. Geophysical Journal International, 214(3):1744–1773, 2018.

[158] Luca Savant, Diego Valsesia, and Enrico Magli. Modeling uncertainty for Gaussian Splatting. arXiv preprint arXiv:2403.18476, 2024.

[159] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. Journal of fluid mechanics, 656:5–28, 2010.

[160] Philip Sedgwick. Spearman's rank correlation coefficient. Bmj, 349, 2014.

[161] Pranab Kumar Sen. Estimates of the regression coefficient based on kendall's tau. Journal of the American statistical association, 63(324):1379–1389, 1968.

[162] Sunando Sengupta, Eric Greveson, Ali Shahrokni, and Philip H. S. Torr. Urban 3D semantic modelling using stereo vision. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 580–585, 2013.

[163] Carmelo Sferrazza, Younggyo Seo, Hao Liu, Youngwoon Lee, and Pieter Abbeel. The power of the senses: Generalizable manipulation from vision and touch through masked multimodal learning. arXiv preprint arXiv:2311.00924, 2023.

[164] Jianxiong Shen, Antonio Agudo, Francesc Moreno-Noguer, and Adria Ruiz. Conditional-Flow NeRF: Accurate 3D modelling with reliable uncertainty quantification. In European Conference on Computer Vision, pages 540–557. Springer, 2022.

[165] Jianxiong Shen, Adria Ruiz, Antonio Agudo, and Francesc Moreno-Noguer. Stochastic neural radiance fields: Quantifying uncertainty in implicit 3d representations. In 2021 International Conference on 3D Vision (3DV), pages 972–981. IEEE, 2021.

[166] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12, pages 746–760. Springer, 2012.

[167] Sergei Abramovich Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. In Doklady Akademii Nauk, volume 148, pages 1042–1045. Russian Academy of Sciences, 1963.

[168] Tomas Soler and John Marshall. Rigorous transformation of variance–covariance matrices of gps-derived coordinates and velocities. GPS Solutions, 6:76–90, 11 2002.

[169] Nitish Srivastava. Improving neural networks with dropout. University of Toronto, 182(566):7, 2013.

[170] Andrew Staniforth and Jean Côté. Semi-lagrangian integration schemes for atmospheric models—a review. Monthly weather review, 119(9):2206–2223, 1991.

[171] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In Proceedings of the IEEE/CVF international conference on computer vision, pages 7262–7272, 2021.

[172] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In Proc. of the International Conference on Intelligent Robot Systems (IROS), Oct. 2012.

[173] Yuxiang Sun, Weixun Zuo, Peng Yun, Hengli Wang, and Ming Liu. Fuseseg: Semantic segmentation of urban scenes based on rgb and thermal data fusion. IEEE Transactions on Automation Science and Engineering, 18(3):1000–1011, 2020.

[174] Niko Sünderhauf, Jad Abou-Chakra, and Dimity Miller. Density-aware NeRF ensembles: Quantifying predictive uncertainty in neural radiance fields. In 2023 IEEE International Conference on Robotics and Automation (ICRA), pages 9370–9376. IEEE, 2023.

[175] Niko Sünderhauf, Trung T Pham, Yasir Latif, Michael Milford, and Ian Reid. Meaningful maps with object-oriented semantic mapping. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5079–5085. IEEE, 2017.

[176] Richard Szeliski. Bayesian modeling of uncertainty in low-level vision. International Journal of Computer Vision, 5(3):271–301, 1990.

[177] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In Proceedings of the European Conference on Computer Vision, 2020.

[178] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In Proceedings of the European Conference on Computer Vision, page 685–702, Berlin, Heidelberg, 2020. Springer-Verlag.

[179] Vladimir Temlyakov and Tino Ullrich. Hyperbolic cross approximation. Springer, 2016.

[180] Vidar Thomée. Galerkin finite element methods for parabolic problems, volume 25. Springer Science & Business Media, 2007.

[181] Sebastian Thrun. Particle filters in robotics. In UAI, volume 2, pages 511–518. Citeseer, 2002.

[182] Sebastian Thrun. Probabilistic robotics. Communications of the ACM, 45(3):52–57, 2002.

[183] Clifford A Truesdell. A First Course in Rational Continuum Mechanics V1. Academic Press, 1992.

[184] Stephen Tu. The dirichlet-multinomial and dirichlet-categorical models for bayesian inference. Computer Science Division, UC Berkeley, 2, 2014.

[185] Aishwarya Unnikrishnan, Joey Wilson, Lu Gan, Andrew Capodieci, Paramsothy Jayakumar, Kira Barton, and Maani Ghaffari. Dynamic semantic occupancy mapping using 3D scene flow and closed-form Bayesian inference. IEEE Access, 10:97954–97970, 2022.

[186] Paul Upchurch* and Ransen Niu*. A dense material segmentation dataset for indoor and outdoor scene parsing. In ECCV, 2022.

[187] SM Utam and J von Neumann. On combination of stochastic and deterministic processes. preliminary report. Bull. Amer. Math. Soe, 53:1120, 1947.

[188] Abhinav Valada, Rohit Mohan, and Wolfram Burgard. Self-supervised model adaptation for multimodal semantic segmentation. International Journal of Computer Vision, 128(5), 2020.

[189] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[190] William R Vega-Brown, Marek Doniec, and Nicholas G Roy. Nonparametric Bayesian inference on multivariate exponential families. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, Proceedings of the Advances in Neural Information Processing Systems Conference, volume 27, 2014.

[191] William R Vega-Brown, Marek Doniec, and Nicholas G Roy. Nonparametric Bayesian inference on multivariate exponential families. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, Proceedings of the Advances in Neural Information Processing Systems Conference, volume 27. Curran Associates, Inc., 2014.

[192] Eric A Wan and Alex T Nelson. Dual extended kalman filter methods. Kalman filtering and neural networks, pages 123–173, 2001.

[193] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Towards unified depth and semantic prediction from a single image. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2800–2809, 2015.

[194] Shifeng Wang. Road Terrain Classification Technology for Autonomous Vehicle. Springer Singapore, 01 2019.

[195] Shifeng Wang. Road terrain classification technology for autonomous vehicle. Springer, 2019.

[196] Zixuan Wang, Qi Tang, Wei Guo, and Yingda Cheng. Sparse grid discontinuous galerkin methods for high-dimensional elliptic equations. Journal of Computational Physics, 314:244–263, 2016.

[197] Jake Welde, Matthew D Kvalheim, and Vijay Kumar. The role of symmetry in constructing geometric flat outputs for free-flying robotic systems. In 2023 IEEE International Conference on Robotics and Automation (ICRA), pages 12247–12253. IEEE, 2023.

[198] Joey Wilson, Yuewei Fu, Joshua Friesen, Parker Ewen, Andrew Capodieci, Paramsothy Jayakumar, Kira Barton, and Maani Ghaffari. Convbki: Real-time probabilistic semantic mapping network with quantifiable uncertainty. Transactions on Robotics, 2024.

[199] Joey Wilson, Yuewei Fu, Arthur Zhang, Jingyu Song, Andrew Capodieci, Paramsothy Jayakumar, Kira Barton, and Maani Ghaffari. Convolutional Bayesian Kernel Inference for 3D Semantic Mapping. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 8364–8370, 2023.

[200] Joey Wilson, Jingyu Song, Yuewei Fu, Arthur Zhang, Andrew Capodieci, Paramsothy Jayakumar, Kira Barton, and Maani Ghaffari. MotionSC: Data Set and Network for Real-Time Semantic Mapping in Dynamic Environments. IEEE Robot. Autom. Letter., 7(3):8439–8446, 2022.

[201] Joey Wilson, Ruihan Xu, Yile Sun, Parker Ewen, Minghan Zhu, Kira Barton, and Maani Ghaffari. Latent bki: Open-dictionary continuous mapping in visual-language latent spaces with quantifiable uncertainty. arXiv preprint arXiv:2410.11783, 2024.

[202] Yuefei Wu, Bin Shi, Bo Dong, Qinghua Zheng, and Hua Wei. The evidence contraction issue in deep evidential regression: Discussion and solution. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, pages 21726–21734, 2024.

[203] Yu Xiang and Dieter Fox. Da-rnn: Semantic mapping with data associated recurrent neural networks. arXiv preprint arXiv:1703.03098, 2017.

[204] Xuan Xiao, Jiahang Liu, Zhipeng Wang, Yanmin Zhou, Yong Qi, Qian Cheng, Bin He, and Shuo Jiang. Robot learning in the era of foundation models: A survey. arXiv preprint arXiv:2311.14379, 2023.

[205] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. Advances in Neural Information Processing Systems, 34:12077–12090, 2021.

[206] Xu Yan, Jiantao Gao, Chaoda Zheng, Chao Zheng, Ruimao Zhang, Shuguang Cui, and Zhen Li. 2dpass: 2d priors assisted semantic segmentation on lidar point clouds. In Proceedings of the European Conference on Computer Vision, pages 677–695. Springer, 2022.

[207] Shichao Yang, Yulan Huang, and Sebastian Scherer. Semantic 3D occupancy mapping through efficient high order CRFs. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 590–597, 09 2017.

[208] Yu Yao, Ella Atkins, Matthew Johnson Roberson, Ram Vasudevan, and Xiaoxiao Du. Coupling intent and action for pedestrian crossing behavior prediction. arXiv preprint arXiv:2105.04133, 2021.

[209] Fisher Yu and Vladlen Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. In Proceedings of the International Conference on Learning Representations, 2016.

[210] Bohao Zhang, Daniel Haugk, and Ram Vasudevan. System identification for constrained robots. arXiv preprint arXiv:2408.08830, 2024.

[211] Bohao Zhang and Ram Vasudevan. Rapid and robust trajectory optimization for humanoids. arXiv preprint arXiv:2409.00303, 2024.

[212] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In Conference on Computer Vision and Pattern Recognition, June 2018.

[213] Juyong Zhang, Bailin Deng, Zishun Liu, Giuseppe Patanè, Sofien Bouaziz, Kai Hormann, and Ligang Liu. Local barycentric coordinates. ACM Transactions on Graphics, 33(6):1–12, 2014.

[214] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. International Journal on Computer Vision, 2018.

[215] Jacek Zienkiewicz, Akis Tsiotsios, Andrew Davison, and Stefan Leutenegger. Monocular, real-time surface reconstruction using dynamic level of detail. In International Conference on 3D Vision, pages 37–46, 2016.