Learning Representations for Safe Autonomous Movement

William Qi

CMU-RI-TR-20-36

August 5, 2020

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Deva Ramanan (Chair)
David Held
Adithya Murali

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

Abstract

Mobile robots have become an increasingly common presence in our homes and on our roads. To move safely within these shared spaces, autonomous agents must understand how other dynamic actors behave and how such behavior influences the *navigability* of the surrounding scene. Towards this goal, we propose two data-driven methods that learn representations from large, self-supervised corpora of data for 1) motion planning in hazardous, dynamic environments and 2) motion forecasting in autonomous driving settings.

The first method - *active affordance learning* (A2L) - advocates for a modular approach to autonomous navigation that combines learned spatial representations with traditional geometry-based maps and classical planning algorithms. By learning to predict a spatial affordance map (that encodes what parts of a scene are navigable) throug‘h active self-supervised experience gathering, we show that A2L is more sample efficient, generalizable, and interpretable than existing state-of-the-art reinforcement learning-based (RL) approaches.

The second method - *what-if motion prediction* (WIMP) - proposes a recurrent graph-based attentional approach for data-driven trajectory forecasting in autonomous driving settings. To the best of our knowledge, this is the first approach to demonstrate counterfactual motion forecasting based on topological queries such as map-based goals and social contexts. We demonstrate the benefits of our proposed framework on the challenging Argoverse vehicle motion forecasting dataset, outperforming previous birds-eye-view (BEV) representation-based methods and setting a new benchmark for prediction quality.

Acknowledgments

I first thank my advisor - Deva Ramanan - for his counsel, support, and patience over the last two years. Under his guidance, I have been afforded invaluable opportunities that have helped me to both expand my horizons and grow as a researcher. I am also grateful to Ravi Teja Mullapudi and Andrew Hartnett for their mentorship; thank you for sharing your knowledge and showing me the ropes. In addition, I would like to thank my committee members, David Held and Adithya Murali, for their time, feedback, and thoughtful questions.

I would also like to to recognize the contributions of my colleagues at Argo AI and within Deva's lab, whose diverse perspectives have helped to shape this work. In particular, I'd like to thank Siddhesh Khandelwal, Jagjeet Singh, Peiyun Hu, and Martin Li for all the interesting discussions we've had.

Lastly, I am incredibly grateful to my family and friends, who have always been with me every step of the way.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Mobile robots, previously confined to restricted-access industrial spaces such as research labs and factories, have become an increasingly common presence in everyday life. Performing routine tasks such as cleaning and driving, autonomous agents now co-habit spaces once exclusively occupied by humans, such as homes and roads.

Operating in close proximity to other dynamic actors such as cars and pedestrians, these agents are expected to move in a predictable, interpretable, and socially compliant manner. As dynamic actors move around the environment, the first and second-order effects of their actions may also introduce additional constraints on agent movement. Effective autonomous agents must be able to capture and plan with regard for these changes in *navigability*, so as to ensure the safety and comfort of all those who share the surrounding environment.

In real-world settings such as autonomous driving, this is a difficult problem to solve. Agents are charged with the task of reaching assigned goals, while simultaneously maintaining acceptable margins of safety during navigation. Prevalent systems often tackle this problem with a combination of static map-building (using simultaneous localization and mapping (SLAM) frameworks) and rule-based planning, relying on frequent updates and rapid re-planning to account for dynamic changes within the surrounding environment. Although practical, use of such *reactive* systems may result in undesirable behavior (such as jerky motion) as agents respond to new information in real-time; this negatively impacts predictability and passenger comfort.

To address this issue, agent motion planning must account for future actions likely to be taken by nearby actors. This can be achieved by *forecasting* the future states of relevant actors, such that non-conflicting ego-trajectories are generated by the planner. This can be challenging, as actor state is only partially observable and future motion is fundamentally multi-modal. In addition, even though a large number of actors may be present in the nearby environment, only a small subset of the extremely large prediction space may be relevant to motion planning; this presents challenges for computational efficiency.

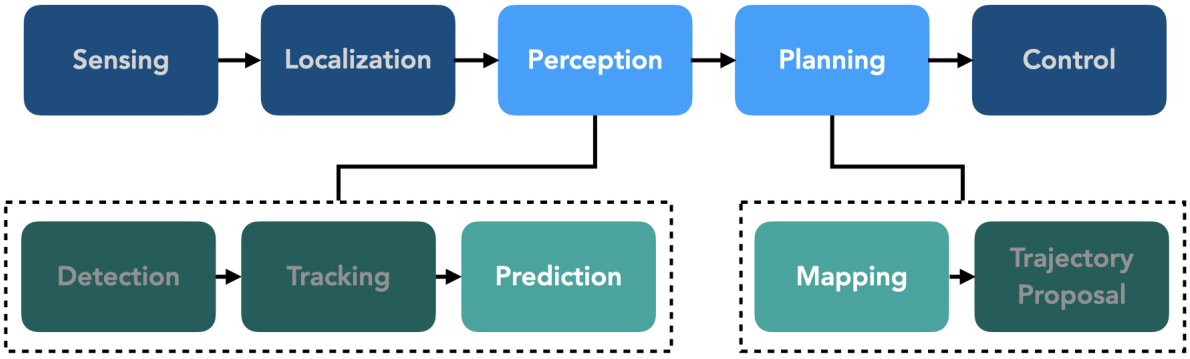

Figure 1.1: Overview of the generalized autonomy stack for mobile robots, with five distinct modules: sensing, localization, perception, planning, and control. In this thesis, we focus primarily on perception and planning, which each can be broken down further into distinct submodules. Of these, we propose and demonstrate improvements for prediction and mapping, which contribute towards the goal of safer autonomous movement.

In the context of prediction and planning, deep neural network-based frameworks [29, 43, 79] have been shown to be particularly effective, learning generalizable representations for scene context that are optimized end-to-end with task-specific losses. Although data hungry, the proliferation of large-scale autonomous vehicle testing [11, 15] and emergence of realistic simulation environments [64, 77] have provided opportunities to obtain labelled data at minimal supervision cost. This motivates a data-driven approach for motion prediction and motion planning - where large-scale collections of data are leveraged to learn representations that capture the dynamic, geometric, semantic, and social attributes of scene context.

1.2 Scope and Approach

In this thesis, we focus on two important and closely-related components of the mobile robot autonomy stack (Fig. 1.1) - motion planning and motion prediction . Approaching each of these tasks from a data-driven, learning-first perspective, we incorporate structured inductive biases to augment existing approaches with learned spatio(temporal) representations that capture complexities within real-world, multi-actor shared spaces (such as homes and streets). Conditioning on both the static scene context and observed actions of other scene participants, these representations are designed to explicitly incorporate the dynamic, geometric, semantic, and social constraints that shape how mobile robots can and should move within our shared spaces. We propose two methods that incorporate such learned representations, targeted at 1) safe motion planning within hazardous, dynamic environments and 2) multi-modal actor motion forecasting for autonomous driving. Empirically, we demonstrate that such learning-based methods are generalizable, performant, and indeed practical for deployment on real-world autonomous systems.

The first method - *active affordance learning* (A2L) - advocates for a modular approach to autonomous navigation that combines learned spatial representations with traditional geometry-

based maps and classical planning algorithms. By learning to predict a spatial affordance map (that encodes what parts of a scene are navigable) through active self-supervised experience gathering, we show that A2L is more sample efficient, generalizable, and interpretable than existing state-of-the-art reinforcement learning-based (RL) approaches. Evaluating in simulation, we also show that learned affordance maps can be used to augment traditional approaches for both autonomous exploration and navigation, providing significant improvements in performance.

The second method - *what-if motion prediction* (WIMP) - proposes a recurrent graph-based attentional approach for data-driven vehicle trajectory forecasting in autonomous driving settings. In contrast to recent state-of-the-art learning-based architectures that generate predictions without regard to the autonomous vehicle's (AV's) intended motion plan, WIMP features interpretable geometric (actor-lane) and social (actor-actor) relationships that support injection of counterfactual geometric goals and social contexts. By designing our approach to efficiently support counterfactual reasoning, we frame prediction as one key component of the *combined* motion prediction-planning loop; this provides additional contextual information, which can be used to generate more relevant predictions. Empirically, we show that our model can produce diverse, multi-modal trajectories conditioned on hypothetical or "what-if" road lane connectivity and multi-actor interactions. We also demonstrate that such an approach could be used in the planning loop to reason about unobserved causes or unlikely futures that are directly relevant to the AV's intended route.

1.3 Contributions and Organization

As highlighted in Section 1.2, this thesis addresses two main topics relating to safe autonomous movement, focusing on motion planning and motion prediction. In Chapter 2, we propose an active affordance learning-based approach for autonomous navigation within dynamic and hazardous environments. Concisely, our primary contributions are as follows:

- We design an agent that learns to predict a spatial affordance map, which encodes what parts of a scene are navigable, using active self-supervised experience gathering.
- We show how learned affordance maps can be combined with classical geometry-based motion planning to avoid dynamic actors and semantic hazards within the environment.
- We empirically demonstrate that affordance-augmented navigation outperforms state-of-the-art RL-based models in hazardous environments, while providing significant advantages in generalizability, interpretability, and sample efficiency.

In Chapter 3, we present the *what-if motion predictor*, a graph-based attentional approach for vehicle trajectory forecasting that integrates tightly with motion planning for autonomous vehicles. In this setting, our primary contributions are as follows:

- We propose a recurrent, graph-based, attentional framework for trajectory forecasting that features interpretable geometric and social relationships.
- To our best knowledge, we are the first to demonstrate counterfactual motion forecasting

based on topological queries such as map-based goals and social contexts.

- We empirically show that the WIMP framework out-performs all previous published approaches on the challenging Argoverse vehicle motion forecasting dataset.

Finally, Chapter 4 summarizes the contributions of this thesis and sets direction for how affordance learning and trajectory forecasting can be extended and combined in future work. We anticipate that further research in this direction will lead to significant improvements in the performance of real-world autonomous navigation system within multi-actor spaces.

Chapter 2

Learning to Move with Affordance Maps

2.1 Motivation

The ability to explore and navigate within a physical space is a fundamental requirement for virtually any mobile autonomous agent, from household robotic vacuums to autonomous vehicles. Traditional approaches for navigation and exploration rely on simultaneous localization and mapping (SLAM) methods to recover scene geometry, producing an explicit geometric map as output. Such maps can be used in conjunction with classic geometric motion planners for exploration and navigation (such as those based on graph search).

However, geometric maps fail to capture dynamic objects within an environment, such as humans, vehicles, or even other autonomous agents. In fact, such dynamic obstacles are intentionally treated as outliers to be ignored when learning a geometric map. However, autonomous agents *must* follow a navigation policy that avoids collisions with dynamic obstacles to ensure safe operation. Moreover, real-world environments also offer a unique set of affordances and semantic constraints specific to each agent: a human-sized agent might fit through a particular door, but a car-sized agent may not; similarly, a bicycle lane may be geometrically free of obstacles, but access is restricted to most agents. Such semantic and behavioral constraints are challenging to encode with classic SLAM.

One promising alternative is *end-to-end* reinforcement learning (RL) of a policy for exploration and navigation. Such approaches have the potential to jointly learn an exploration/navigation planner together with an internal representation that captures both geometric, semantic, and dynamic constraints. However, such techniques suffer from well-known challenges common to RL such as high sample complexity (because reward signals tend to be sparse), difficulty in generalization to novel environments (due to overfitting), and lack of interpretability.

We advocate a hybrid approach that combines the best of both worlds. Rather than end-to-end learning of both a spatial representation and exploration policy, we apply learning only “as needed". Specifically, we employ off-the-shelf planners, but augment the classic geometric map with a *spatial affordance map* that encodes where the agent can safely move. Crucially, the

affordance map is learned through self-supervised interaction with the environment. For example, our agent can *discover* that spatial regions with wet-looking floors are non-navigable and that spatial regions that recently contained human-like visual signatures should be avoided with a large margin of safety. Evaluating on an exploration-based task, we demonstrate that affordance map-based approaches are far more sample-efficient, generalizable, and interpretable than current RL-based methods.

Even though we believe our problem formulation to be rather practical and common, evaluation is challenging in both the physical world and virtual simulators. It it notoriously difficult to evaluate real-world autonomous agents over a large and diverse set of environments. Moreover, many realistic simulators for navigation and exploration assume a static environment [63, 74, 77]. We opt for first-person game-based simulators that populate virtual worlds with dynamic actors. Specifically, we evaluate exploration and navigation policies in VizDoom [76], a popular platform for RL research. We demonstrate that affordance maps, when combined with classic planners, dramatically outperform traditional geometric methods by 60% and state-of-the-art RL approaches by 70% in the exploration task. Additionally, we demonstrate that by combining active learning and affordance maps with geometry, navigation performance improves by up to 55% in the presence of hazards. However, a significant gap still remains between human and autonomous performance, indicating the difficulty of these tasks even in the relatively simple setting of a simulated world.

2.2 Related Work

Navigation in Classical Robotics. Navigation has classically been framed as a geometry problem decomposed into two parts: mapping and path planning. Inputs from cameras and sensors such as LiDARs are used to estimate a geometric representation of the world through SLAM (or structure from motion) techniques [10, 71]. This geometric representation is used to derive a map of traversability, encoding the likelihood of collision with any of the inferred geometry. Such a map of traversability can be used with path planning algorithms [12, 37, 42] to compute collision-free paths to desired goal locations. Navigation applications can be built upon these two primitives. For example, exploration of novel environments can be undertaken by sampling point goals in currently unknown space, planning paths to these point goals, and incrementally building a map using the sensor measurements along the way (also known as frontier-based exploration [78]). Such an approach for exploration has proven to be highly effective, besting even recent RL-based techniques in static environments [17], while relying on classical planning [27].

Semantics and Learning for Navigation. Taking a purely geometric approach to navigation is very effective when the underlying problem is indeed geometric, such as when the environment is static or when traversability is determined entirely by geometry. However, an entirely geometric treatment can be sub-optimal in situations where semantic information can provide additional cues for navigation (such as emergency exit signs). These considerations have motivated study on semantic SLAM [41], that seeks to associate semantics with maps [8, 49], speed up map building through active search [44], or factor out dynamic objects [6].

In a similar vein, a number of recent works also investigate the use of learning to solve navigation tasks in an end-to-end manner [26, 30, 52, 66, 82], built upon the theory that an agent can automatically learn about semantic regularities by directly interacting with the environment. Semantics have also been used as intermediate representations to transfer between simulation and the real world [53]. While such use of learning is promising, experiments in past work have focused only on semantics associated with static maps. Instead, we investigate the role of semantics in dynamic environments, and in scenarios where the notion of affordance goes beyond simple geometric occupancy.

Another recent approach [51] introduces a method of learning generalized spatial representations for both exploration and navigation, employing an attention-based generative model to reconstruct geometric observations. Planning for navigation occurs in belief space, in contrast to the metric cost maps (incorporating both semantics and geometry) used in our work.

Hybrid Navigation Policies. While learning-based methods leverage semantic cues, training such policies can be sample inefficient. This has motivated the pursuit of hybrid policy architectures that combine learning with geometric reasoning [7, 30] or known robot dynamic models [4, 36, 53]. Our work also presents a hybrid approach, but investigates fusion of a learned mapper with analytic path planning.

Self-Supervised Learning. Recent work in robotics has sought to employ self-supervised learning [56] as an alternative to end-to-end reward-based learning. [32] and [9] employ passive cross-modal self-supervision to learn navigability (from stereo to monocular images, and from LiDAR to monocular images, respectively). In contrast, we learn through active interaction with the environment. Thus, our work is most similar to that of [28], though we learn *dense* traversibility predictions for long range path-planning, rather than short-range predictions for collision avoidance.

Navigation in Dynamic Environments. Finally, a number of other works develop specialized techniques for navigation in dynamic environments, by building explicit models for other agents' dynamics [16, 40, 54]. In contrast, by generalizing our definition of traversability beyond geometry alone, we can automatically capture the dynamics of other agents implicitly and jointly with other environmental features.

2.3 Approach

Our goal is to build an agent that can efficiently explore and navigate within novel environments populated with other dynamic actors, while respecting the semantic constraints of the environment. The scenario we consider is a mobile agent capable of executing basic movement macro-actions. The agent is equipped with a RGBD camera and some form of proprioceptive feedback indicating well-being (e.g bump sensor, wheel slip, game damage). We assume that the agent is localized using noisy odometry and that depth sensing is also imperfect and noisy. At test time, the agent is initialized within a novel environment containing an unknown number of dynamic and environmental hazards. Furthermore, we assume that the exact dimensions of the agent and nature of affordances provided by entities within the environment are not initially known.

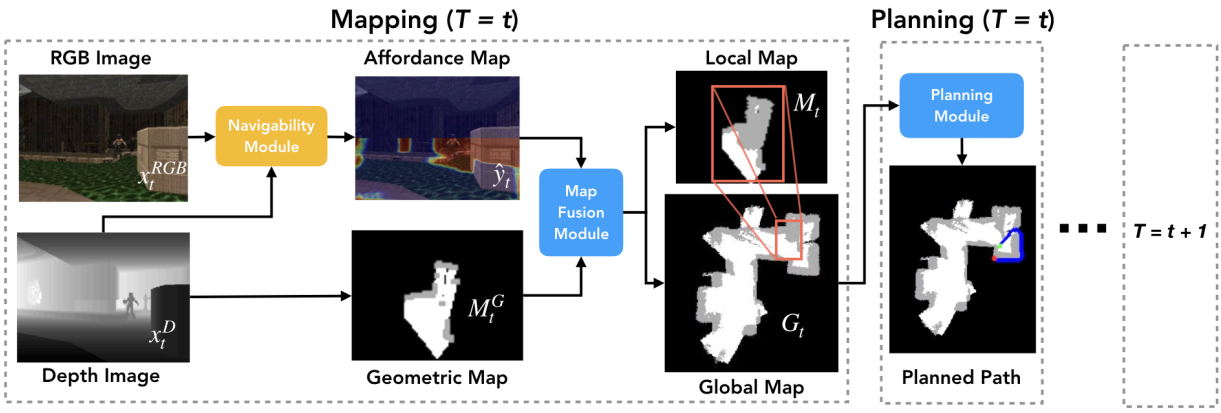

Figure 2.1: Overview of our proposed architecture for navigation. RGBD inputs x_t are used to predict affordance maps $\hat{y}_t$ and transformed into egocentric navigability maps M_t that incorporate both geometric and semantic information. In the example shown, M_t is labelled as non-navigable in regions near the monster. A running estimate of the current position at each time step is maintained and used to update a global, allocentric map of navigability G_t that enables safe and efficient planning.

We propose a modular approach to tackle this problem, adopting a classical pipeline of map building and path planning. Figure 2.1 shows an overview of this pipeline, which builds a navigability map using both geometric and semantic information, as opposed to traditional methods that rely on geometry alone. Our main contribution, shown in Figure 2.2, is a method for predicting which parts of a scene are navigable by actively leveraging the feedback sensor to generate partially-labeled training examples. We then use the labeled samples to train a model which predicts a per-pixel affordance map from the agent's viewpoint. At evaluation time, the outputs from the learned module are combined with geometric information from the depth sensor to build egocentric and allocentric representations that capture both semantic and geometric constraints. The fused representations can then be used for exploration/navigation by employing traditional path planning techniques, enabling safe movement even within dynamic and hazardous environments.

2.3.1 Navigability Module

Given a scene representation x captured by a RGBD camera, our goal is to train a module π that labels each pixel with a binary affordance value, describing whether the corresponding position is a valid space for the agent to occupy and forming a segmentation map of "navigability" y. We can encode this understanding of the environment by training an image segmentation model in a supervised fashion. However, training such a model requires a set of labeled training images $D = [(x_1, y_1), ...(x_n, y_n)]$ where each pixel is annotated for navigability. Traditionally, obtaining such a set of labels has required dense annotation by an oracle [18], at a cost that scales linearly with the amount of data labeled. These properties have generally limited applications to domains captured by large segmentation datasets [46, 81] that have been curated using hundreds of hours of human annotation time. We address this problem by employing a self-supervised approach to generate partially labeled examples $\tilde{y}$, in place of oracle annotation.

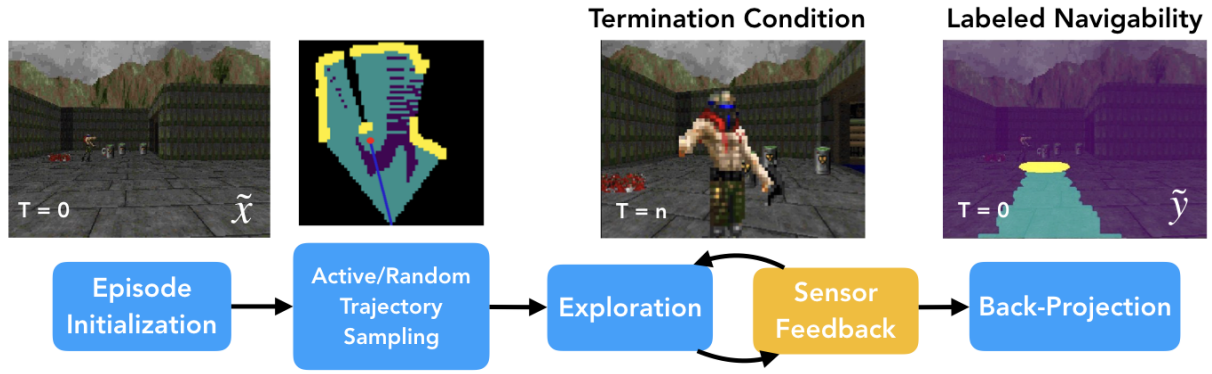

Figure 2.2: Overview of self-supervised labeling for navigability training pairs $(\tilde{x}, \tilde{y})$. The agent performs a series of walks along random or planned trajectories within the environment. Affordance information collected from each walk is back-projected onto pixel-level labels in the agent's POV from previous time steps. Sampling over a variety of maps allows for the collection of a visually and semantically diverse set of examples $\tilde{D}$ that can be used to train a navigability module π. This figure illustrates the generation of a negative example, with the agent contacting a dynamic hazard.

Self-Supervision. We generate labeled affordance data in a self-supervised manner through continuous interactive exploration by the agent; this algorithm makes use of RGBD observations x, readings from a feedback sensor s, and a history of actions a_t executed over time. In each episode, the agent is initialized at a random location and orientation within a training environment. The agent selects a nearby point and attempts to navigate towards it. Labeled training data is generated based on whether or not the agent is able to reach this point: every location that the agent successfully traverses during its attempt is marked as navigable, while undesirable locations (e.g. bumping into obstacles, loss of traction, loss in health, getting stuck) are marked as non-navigable. These locations in world space are then back-projected into previous image frames using estimated camera intrinsics, in order to obtain partial segmentation labels (examples of which are visualized in Figure 2.3). Pixels for which there are no positive or negative labels are marked as unknown. A more detailed discussion about the real-world applicability of this approach can be found in Section 2.5.2.

Dense Labels. Backprojection of affordance labels produces a dense set of *pixelwise* labels for observations at past time steps. Importantly, even without spatio-temporal inputs, this enables the training of models which incorporate safety margins to account for motion, as the future position of dynamic actors is encoded within labelled views from the past (discussed further in Section 2.5.1). In contrast, most RL-based methods return only a single sparse scalar reward, which often leads to sample inefficient learning, potentially requiring millions of sampling episodes [82]. Furthermore, our generated labels $\tilde{y}$ are human interpretable, forming a mid-level representation that improves interpretability of actions undertaken by the agent.

Navigability Segmentation. The collected samples $\tilde{D}$ are used to train a segmentation network such as UNet [60], allowing for generalization of sampled knowledge to novel scenarios. A masked loss function $L_{mask} = K \odot L_{BCE}(\hat{y}, y)$ based on binary cross-entropy is employed to

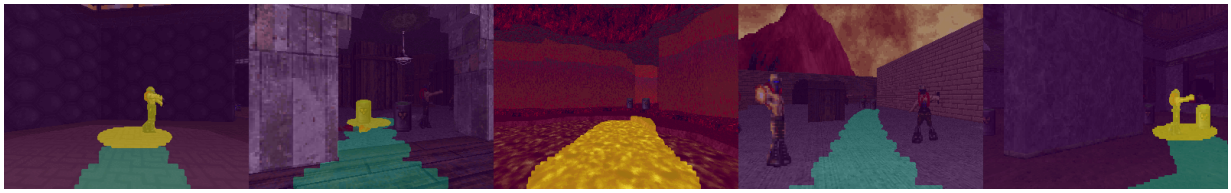

Figure 2.3: Examples of samples labeled through back-projection (navigable area labeled in green, non-navigable in yellow, and unknown in purple). The first three examples show negative examples, labeled by damage from monster, impediment of movement by barrel, and damage taken from environmental hazard respectively. The fourth illustrates successful traversal between monsters and the fifth shows an example collected along a minimum cost path as part of an active learning loop.

ensure that only labeled, non-unknown points K within each example contribute to the loss.

As labeled samples $\tilde{y}$ are generated by painting all pixels within a radius of a specified point in the simulated world as either navigable or non-navigable, we are more "certain" that pixels close by in world space share the same semantic constraints than those farther away. To ensure stability of training, we express this belief in the loss by weighting each pixel's contribution by its inverse Euclidean distance to the closest such point in world space. Given enough training data, the navigability module is capable of generating segmentation maps that closely approximate ground truth navigability, even in previously unseen environments (shown in Figure 2.4).

Active Trajectory Sampling. In order to further improve sample efficiency, we can employ model uncertainty to actively plan paths during sampling episodes so as to maximize label entropy along traversed trajectories. Intuitively, many semantically interesting artifacts (such as environmental hazards) are rare, making it difficult to learn a visual signature. In these cases, sampling can be made more efficient by intentionally *seeking* out such artifacts. This can be achieved by first collecting a small number (n) of samples using random walks and training a seed segmentation model. Using the seed model, we then predict an affordance map $\hat{y}$ during the first step of each subsequent episode and use it to construct a cost map for planning, with values inversely proportional to the prediction uncertainty (defined as the entropy of the predicted softmax distribution over class labels) at each position. Planning and following a minimal-cost path in this space is equivalent to maximization of label entropy, as the agent will attempt to interact most with highly uncertain areas. Once an additional n samples have been actively collected using this strategy, the model is retrained using a mixture of all samples collected so far, and the sample/train loop can be repeated again. We find that active learning *further* increases our sample efficiency, requiring fewer sampling episodes to learn visual signatures for hazards and dynamic actors (example shown in Figure 2.3 rightmost).

2.3.2 Map Construction

While some hazards can only be identified using semantic information, geometry provides an effective and reliable means to identify navigability around large, static, obstacles such as walls. To capture both types of constraints, we augment our predicted semantic maps with additional

Figure 2.4: Examples of affordance maps $\hat{y}$ predicted by the navigability module, showing accurate localization of semantic constraints within the scene. (**Left**) contains dynamic hazards in the form of monsters and (**Right**) contains areas of geometry-affordance mismatch, in the form of barrels shorter than sensor height.

geometric information when constructing the projected navigability cost maps M and G used for planning. As the agent moves around the environment, observed depth images are used to construct local, egocentric occupancy maps at each time step, incorporating only geometric information. By reading depth values from the center scanline of the depth image, projecting into the XY-plane, and marking the corresponding cells as non-navigable, a map of geometric obstacles M_t^G can be obtained. As the exact dimensions and locomotion capabilities of the agent are unknown, only depth values returned by the center scanline are known to be obstacles with certainty.

Map Fusion. Given a pixel-wise affordance map $\hat{y}_t$ obtained from the navigability module and a local, egocentric geometric map M_t^G, the two inputs can be combined using a fusion module $F(\hat{y}_t, M_t^G)$ to form a single local navigation cost map M_t that incorporates both semantic and geometric information. To do so, the segmentation map $\hat{y}_t$ is first projected into the 2D plane using estimated camera intrinsics, forming an egocentric navigability map M_t^S. Cells marked as obstacles by M_t^G are also marked as impassable within M_t, with remaining cells in free space assigned cost values inversely proportional to the confidence of navigability provided by $\hat{y}_t$. Finally, M_t is used to update a global, allocentric map G_t of navigability at the end of each time step.

2.3.3 Planning

Given the global navigability map, path planning can be tackled using classical algorithms such as A*, as all required semantic and geometric information is encoded within the map itself. Additionally, since both M_t and G_t are updated at every time step, dynamic hazards are treated as any other obstacle and can be avoided successfully as long as paths are re-planned at sufficiently high frequency. Our work is agnostic to the choice of planning algorithm and our semantic maps can also be employed with more sophisticated planners, though for simplicity we evaluate using A*.

2.4 Experiments

We perform our evaluation in simulation using VizDoom, as it allows for procedural generation of large, complex 3D maps that contain a variety of dynamic actors and semantic constraints in the form environmental hazards. Although prior work [62] on navigation has also relied on VizDoom, evaluation has been restricted to a small set of hand designed maps without any dynamic actors or semantic constraints. We evaluate the effectiveness of incorporating learned affordance maps to tackle two difficult tasks: novel environment exploration and goal-directed navigation.

2.4.1 Experimental Setup

We conduct our experiments within procedurally-generated VizDoom maps created by the Oblige [2] level generator, which enables construction of training and test maps containing unique, complex, and visually diverse environments. Each generated map is large, containing a variety of dynamic hazards (such as monsters) and environmental hazards (such as lava pools), in addition to static obstacles (such as barrels) and areas where a geometry-affordance mismatch exists (such as ledges lower than sensor height, but beyond the movement capabilities of the agent). We generate a collection of 60 training and 15 test maps and further categorize the 15 test maps as either hazard-dense or hazard-sparse, based on concentration of hazards within the initial exploration area.

Observation and Action Space. We assume that the agent's RGBD camera returns a regular RGB image with a 60° field of view and an approximately-correct depth image that records the 2D Euclidean distance of each pixel from the camera in the XY plane (due to the 2.5D nature of the Doom rendering engine). The feedback sensor returns a scalar value corresponding to the magnitude of damage received by the agent while executing the previous action (some hazards are more dangerous than others). The action space is limited to three motion primitives: move forward, turn left, and turn right; only one action can be executed at each time step. Localization is imperfect and achieved through odometry from noisy measurements, with approximately 2% error.

Environments. Each test map used to evaluate exploration performance is extremely large, containing sequences of irregularly shaped rooms connected by narrow hallways and openings. As it would require upwards of 10,000 time steps even for a skilled human to explore any of these environments, most of the space covered by the evaluated approaches is contained within the initial rooms next to the start point. As such, to clearly illustrate the challenges posed by semantic constraints, we choose to further categorize the test maps as either *hazard-sparse* or *hazard-dense*, based on the likelihood of encountering navigability-restricting hazards within the initial exploration area. Figure 2.5 shows a top-down visualization of the difference in hazard concentration between the two test subsets.

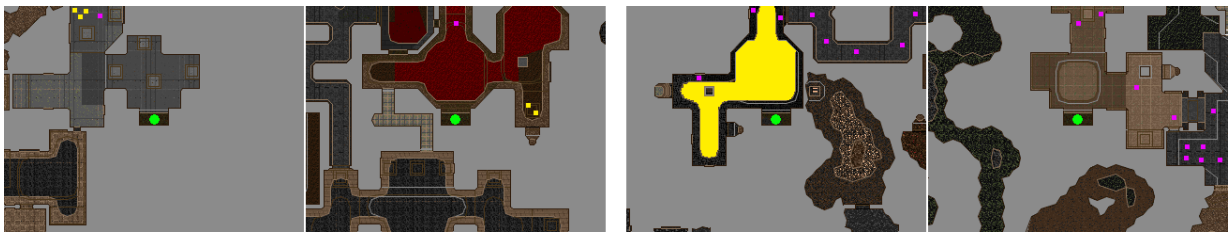

Figure 2.5: Top-down visualizations of initial exploration areas in hazard-sparse **(Left)** and hazard-dense **(Right)** test environments. Agent start position is marked in green, with environmental hazards marked in yellow, and initial locations of dynamic hazards marked in purple. Hazard-dense environments present a significant challenge for autonomous exploration, containing a high concentration of navigability restricting areas that must be avoided successfully.

2.4.2 Sample-Efficient Exploration using Affordance Maps

We quantitatively evaluate exploration performance by measuring the total amount of space observed within a particular environment over time, approximated by the total surface area of the constructed global map. Each episode of evaluation terminates after 2000 time steps or after receiving a total of 100 damage during exploration, whichever occurs first. Agents receive 4 damage per time step when coming into contact with dynamic hazards and 20 damage for environmental hazards.

Baselines We compare our proposed approach against several competitive baselines that range from classical geometry-based approaches to recent RL-based methods. Next, we explain each of these exploration baselines in detail:

1. *Random.* In order to show that both geometry and learning-based approaches set a competitive baseline and perform well above the bar set by random exploration, we evaluate a policy that selects between each of the available actions with uniform probability at each time step.

2. *Frontier-Based Exploration.* As a classical, non-learning baseline, we compare against a variant of frontier-based exploration [25, 78]. This approach relies purely on geometry, updating a global map G_t at every step using the projected scanline observation M_t^G from the current POV. A close-by goal from within the current "frontier region" is selected and a path towards it is re-planned (using A*) every 10 steps as the map is updated. Once the selected goal has been reached or the goal is determined to no longer be reachable, the process is repeated with a newly-selected goal. Although dynamic actors can be localized using geometry alone, they are treated as static obstacles in the cost map, relying on frequent re-planning for collision avoidance.

3. *RL-Based Exploration.* We also compare against a state-of-the-art deep RL-based approach [17] for exploration that is trained using PPO [65] and incorporates both geometric and learned representations. We implement an augmented variant of the method proposed by [17], adding an additional depth map x_t^D to the 3 original inputs: the current RGB observation x_t^{RGB}, a small-scale egocentric crop of G_t, and a large-scale egocentric crop of G_t. To ensure a fair comparison, we evaluate this approach using hyper-parameters identical to those

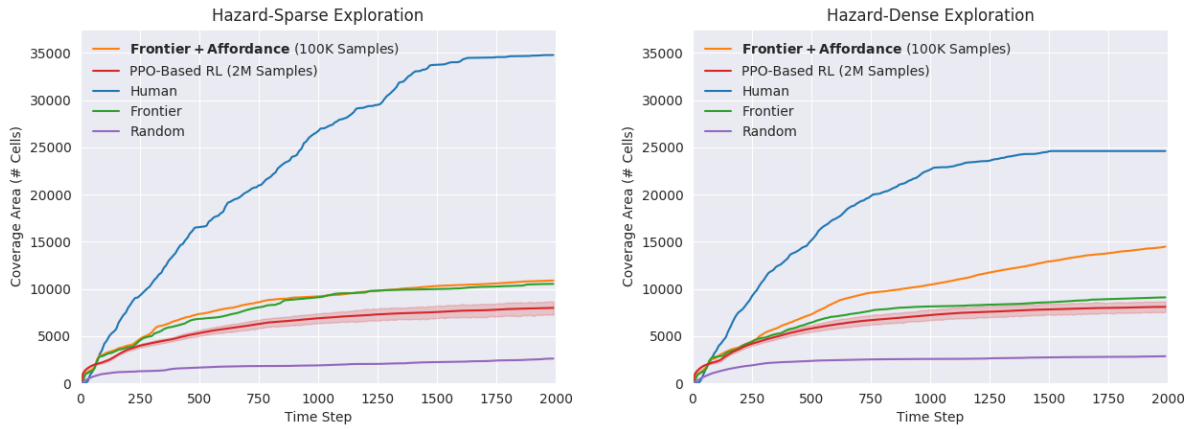

Figure 2.6: Comparison of exploration performance across all evaluated approaches in **(Left)** hazard-dense and **(Right)** hazard-sparse environments, plotted as a function of area observed over time. Both plots report mean performance measured over 5 test episodes; RL results are reported using the best model obtained over 3 randomly-initialized training runs, the shaded area indicates the range of measured values across all test episodes.

proposed by the original authors, maintaining a global map where each grid cell represents an 8×8 game unit area in the VizDoom world. We also introduce a new penalty term used to scale the reward by amount of damage taken in each step (set to 0.02), which encourages the agent to avoid hazardous regions of the map. We report the mean performance obtained by the best model from each of 3 training runs (2M samples each), with access to the full 60 map training set.

4. *Human.* To demonstrate that all autonomous approaches for exploration still have significant room for improvement, we asked human volunteers to explore a sequence of novel environments using a near-identical experimental setup. For ease of use, we adopt a standard Doom control scheme, allowing participants to execute actions for rotation and forward movement concurrently. This provides a slight edge in locomotion compared to other approaches.

Affordance-Augmented Frontier Exploration. To evaluate the efficacy of our proposed representation, we augment the frontier-based approach with semantic navigability maps obtained from affordance predictions; all other components (including goal selection and path planning) are shared with the baseline. We collect approximately 100k total samples across the 60 training maps in a self-supervised manner and train the navigability module for 50 epochs using the collected dataset; a ResNet-18-based [34] UNet [60] architecture is employed for segmentation. Episodic sample goals are selected randomly from within the initial visible area and simple path planning is employed, with the agent always taking a straight line directly towards the goal. Back-projection is performed using game damage as a feedback mechanism, with the size of negative labels corresponding to the magnitude of damage received. At test time, we use estimated camera intrinsics to project output from the navigability module into the 2D plane.

Inside hazard-sparse environments (Figure 2.6 left), agents generally don't encounter hazards

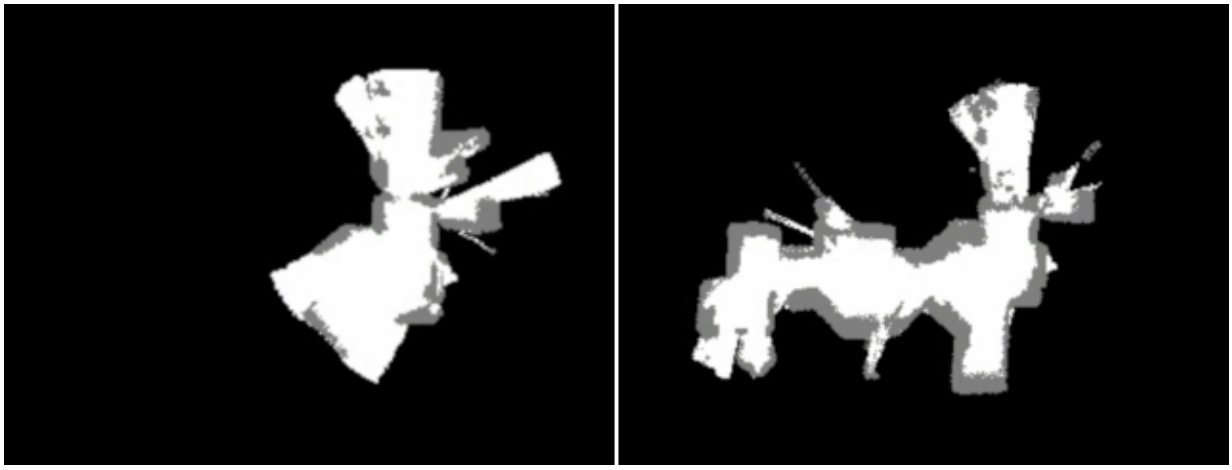

Figure 2.7: Comparison of thresholded global maps constructed by frontier exploration using geometry (**Left**) and affordance-based (**Right**) representations in the same environment. In this setting, semantic representations help the agent take less damage over time, allowing for more area to be explored during each episode.

within the first 2000 time steps, placing increased emphasis on goal selection over hazard avoidance. In this setting, augmenting the frontier-based approach with affordance maps does not provide significant improvements, as in the absence of semantic hazards, the two methods are functionally equivalent. In line with previous work [17], the PPO-based RL approach also fails to beat the frontier baseline, likely due to the heavy emphasis placed on exploration policy. Without taking a high-level representation of the global map as input, it is difficult for a RL-based approach to plan over long time horizons, causing the agent to potentially re-visit areas it has already seen before. Finally, we note that humans are much better at both goal selection and hazard avoidance, managing to explore upwards of $3\times$ more area than the closest autonomous approach.

Successful exploration in hazard-dense environments (Figure 2.6 right) necessitates the ability to identify affordance-restricting hazards, as well as the ability to plan paths that safely navigate around them. In this setting, augmenting the frontier-based approach with affordance maps increases performance by approximately 60%, which is more than 2/3 of the difference between frontier and the random baseline (constructed global maps are compared in Figure 2.7). Qualitatively, we observe that agents using learned affordance maps plan paths that leave a wide margin of safety around observed hazards and spend far less time stuck in areas of geometry-affordance mismatch. Through self-supervised sampling, the navigability module also learns about agent-specific locomotion capabilities, predicting when low ceilings and tall steps may restrict movement. Although RL-based exploration out-performs the frontier baseline in this scenario by learning that proximity to hazards is detrimental to reward maximization, a lack of long term planning still hinders overall exploration performance.

Sample Efficiency. In order to understand the effect of training set size on learned exploration, we measure exploration performance with different amounts of collected samples in the hazard-dense setting, shown in Figure 2.8. After collecting as few as 5000 training samples, the nav-

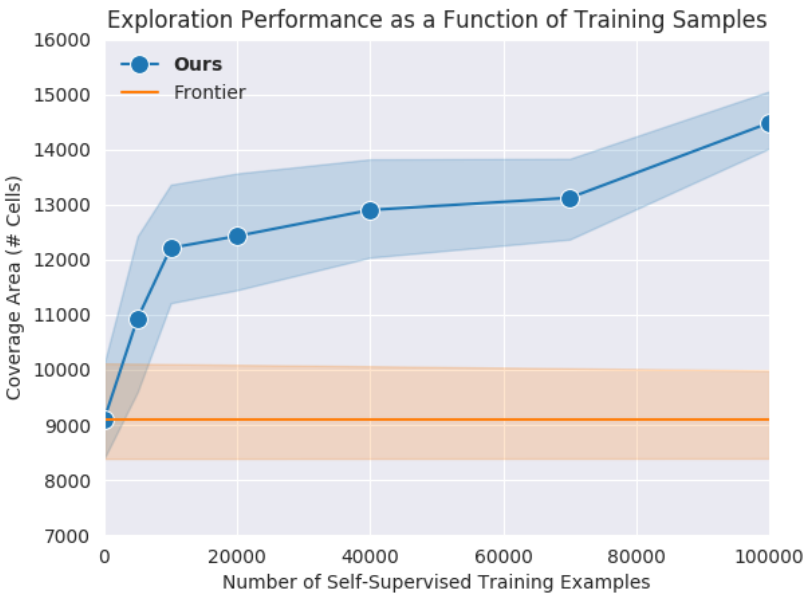

Figure 2.8: Comparison of final exploration coverage achieved by affordance-augmented frontier exploration, trained using varying amounts of self-supervised training data. Plotted lines report mean performance measured over 5 test episodes, while shaded areas indicate the range of values observed during evaluation.

igability module learns to recognize dynamic hazards, allowing for paths to be planned with a margin of safety. As the number of samples collected increases, exploration performance improves as well. However, as one might expect, the relative gain provided by each additional example decreases after a point. Qualitatively, we observe that 10,000 samples provides sufficient diversity to enable accurate localization of common dynamic hazards, while additional examples beyond this point help to improve detection of less commonly observed environmental hazards and accuracy near hazard boundaries. Notably, even after training on 20 times as many samples, RL-based exploration still fails to outperform our approach in this setting, illustrating a clear advantage in sample efficiency.

2.4.3 Goal-Directed Navigation using Active Affordance Map Learning

In order to further demonstrate the applicability and efficacy of affordance-based representations, we set up a series of 15 navigation trials, one for each map in the test set. Within each trial, the agent begins at a fixed start point and is tasked with navigating to an end goal (specified in relative coordinates) in under 1000 time steps, while minimizing the amount of damage taken along the way. Each trial is designed to be difficult, with numerous hazards and obstacles separating the start and end points, presenting a challenge even for skilled humans; none of the human participants were able to complete all 15 trials successfully without taking any damage.

Geometry-Based Navigation Baseline

We implement a classical geometry-based navigation baseline using a modified variant of the planning and locomotion modules employed for frontier-based exploration. Global navigability maps used for planning are constructed by averaging values obtained from local maps over multiple time steps, allowing for increased temporal stability and robustness to sensor and localization noise. A simple A*-based algorithm is then employed for planning, treating the value of each cell in the global navigability map as the cost of traversing through a particular location in the environment.

In this setup, dynamic actors are treated as static obstacles within the cost map, an assumption that holds true as long as re-planning is invoked at a sufficiently high frequency. In order to evaluate the effect of re-planning frequency on navigation performance, we also evaluate a variant of the baseline approach that re-plans at $10\times$ the frequency (every step instead of every 10 steps) and observe that this results in a small improvement in navigation trial success rate, largely attributed to the reduction in latency required to respond to environmental dynamics.

Qualitatively, we observe that most failures in the baseline occur when the agent attempts to path through an obstacle lower than sensor height, causing the agent to become stuck as it continually tries to path through a cell registered as free space in the geometric map. The second most common scenario that leads to degraded performance is failure to avoid dynamic hazards, causing agents to collide with monsters as they attempt to follow a nearby path to the goal.

Affordance-Augmented Navigation

In the evaluated setting, we show that by adding semantic information obtained from affordance maps, it is possible to improve navigation performance significantly, even when employing simple geometry-based approaches that plan using A*. By introducing a navigability module trained on 100k collected samples to generate cost maps for planning, we observe a 45% improvement in overall navigation success rate, with improvements of 25% observed even when using a model trained on a dataset just one fifth the size. Even when the re-planning frequency is increased 10-fold, such that observed dynamic hazards can be treated as static obstacles more accurately, the baseline still fails to beat the affordance-augmented variant.

Additionally, we compare against results obtained by a PPO-based RL model, which is trained similarly to its counterpart discussed in Section 2.4.2. In order to reduce the difficulty of planning over long time horizons, we provide the model with a sequence of waypoints (extracted from the best-performing human trajectory) as an additional input, which are used as local intermediate goals that converge towards a faraway global goal. However, we observe that even with this augmented set of inputs, the RL-based approach still fails to beat any of the affordance-based methods, echoing results observed in the exploration experiments.

We also explore how active learning can be used to further improve the efficiency of self-supervised learning, by evaluating two additional models trained on samples collected from actively-planned trajectories. We show that using just 40% of the data, models employing active data collection outperform those trained using random samples alone. At the 100k total sample

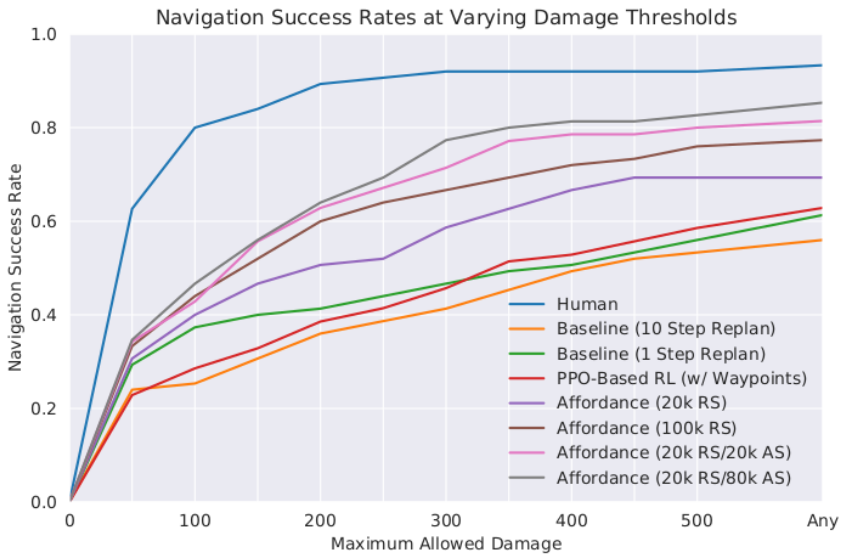

Figure 2.9: Comparison of navigation performance across all evaluated approaches, plotted as a function of success rate vs. maximum amount of damage permitted per trial (mean results over 5 test runs reported).

mark, we observe that actively sampled models out-perform their randomly sampled counterparts by more than 10%. The procedure we follow is to first train a seed model using 20k random samples, before collecting an additional 20k samples actively and re-training on the combined dataset to obtain an improved model. We repeat the active sample/train loop for 3 additional iterations, building a dataset with a total size of 100k samples.

Results obtained by the affordance-augmented navigation agent, along with comparisons to the baseline, are summarized in Figure 2.9; examples of actively-planned trajectories are shown in Figure 2.10. Qualitatively, we observe that active trajectory sampling significantly improves temporal stability and prediction accuracy along hazard and obstacle boundaries (shown in Figure 2.11). These properties enable more efficient path planning, allowing the agent to move safely with tighter margins around identified hazards.

2.5 Discussion

In this chapter, we have described a learnable approach for exploration and navigation within novel environments. Like RL-based policies, our approach learns to exploit semantic, dynamic, and even behavioural properties of the novel environment when navigating (which are difficult to capture using geometry alone). But unlike traditional RL, our approach is made sample-efficient and interpretable by way of a spatial affordance map, a novel representation that is interactively-trained so as to be useful for navigation with off-the-shelf planners.

In the remainder of this section, we 1) discuss how active affordance learning implicitly captures dynamic behavior without temporal actor tracking and 2) explain how our proposed approach can

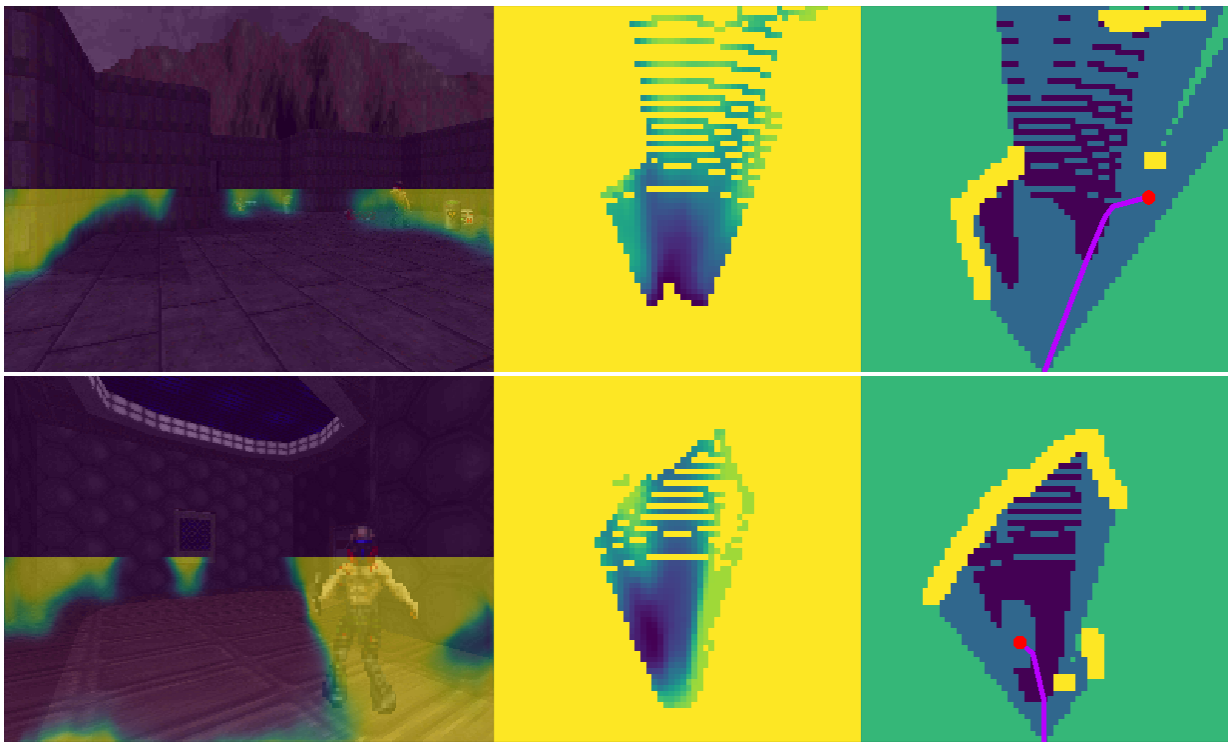

Figure 2.10: Examples of actively-planned trajectories that maximize label entropy along sampled locations. (**Left**) shows predicted affordances, (**Middle**) shows the projected confidence map, and (**Right**) shows the cost map used to plan the optimal path.

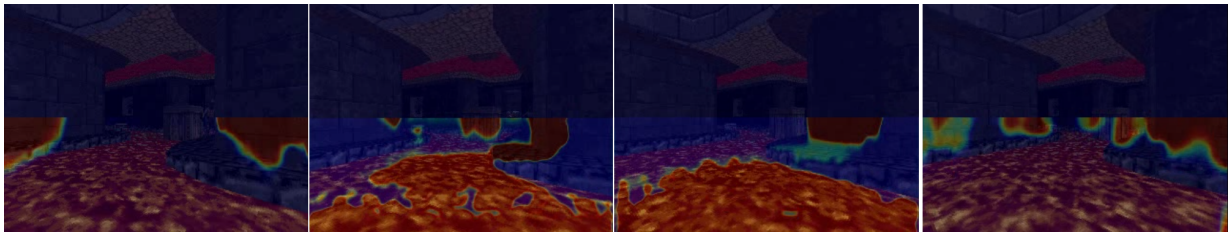

Figure 2.11: Comparison of affordance maps generated by models trained using datasets containing (a) 20k random samples, (b) 20k random samples + 40k active samples, (c) 20k random samples + 80k active samples, and (d) 100k random samples. Employing active learning allows models to effectively identify and localize regions containing rare environmental hazards, a feat that is difficult to achieve using random samples alone.

be applied in real-world robotics settings, where the cost associated with naive "trial and error" approaches may be unacceptably high. Finally, we provide guidance on reproducibility and extensibility, releasing an open-source implementation of the active affordance learning framework as described in this chapter.

2.5.1 Capturing Dynamic Behavior

In order to better understand how dynamic behavior is captured using our affordance-labeling approach, we pose a scenario where a dynamic actor moves from point A to point B, and collides with the agent at point B. In this scenario, all observations of point B (including those collected pre-collision) will be labelled as hazardous, potentially mapping to an image region near the dynamic actor rather than the actor itself. We will next describe one approach for explicitly modeling such moving obstacles, and then justify why our current approach implicitly captures such dynamics.

Explicit Approach. In principle, our self-supervised labeling system can be modified to replace naive back-projection with an explicit image-based tracker (keeping all other components fixed). Essentially, labeled patches can be tracked backwards from the final timestep at which they are identified as hazardous (since those prior visual observations are available at sample time) to obtain their precise image coordinates when backprojecting to prior timesteps.

Implicit Approach. Even without image-based tracking, our pipeline implicitly learns to generate larger safety margins for visual signatures that have been associated with dynamic behavior. Essentially, our system learns to avoid regions that are spatially close to dynamic actors (as seen in Figure 2.12). Such notions of semantic-specific safety margins (e.g., autonomous systems should use larger safety margins for pedestrians vs. roadside trash cans) are typically hand-coded in current systems, but these emerge naturally from our learning-based approach. As we found success with an implicit encoding of dynamics, we did not experiment with explicit encodings, but this would certainly make for interesting future work.

2.5.2 Real-World Applicability

During the sampling stage of our proposed method, we employ a "trial and error"-style approach (similar to RL-based methods) that could lead to potentially hazardous situations in real-world robotics settings if deployed in certain configurations. However, we argue that this is not an unreasonable way of collecting data and that there exist both common and practical solutions for risk mitigation that have already been widely deployed in the real-world.

Framing the current state of self-driving research within the context of our work, we can view all autonomous vehicles today as being within the "sampling" stage of a long-term active learning loop that ultimately aims to enable L4 autonomy. Almost every one of these vehicles on public roads today is equipped with one, if not multiple safety operators who are responsible for disengaging autonomy and "taking over" when the system fails to operate within defined bounds for safety and comfort. Moreover, each of these "takeover" scenarios is logged and used to improve the underlying models in future iterations of this learning loop [23]. Indeed, in this scenario, the safety operator serves the purpose of the "feedback sensor" and can ultimately be removed at "test time", once the autonomous driving model has been deemed safe.

In less safety critical scenarios, such as closed course or small-scale testing, the role of the safety driver could be replaced with some form of high-frequency, high-resolution sensing such as multiple short-range LIDARs. These feedback sensors can be used to help the robot avoid

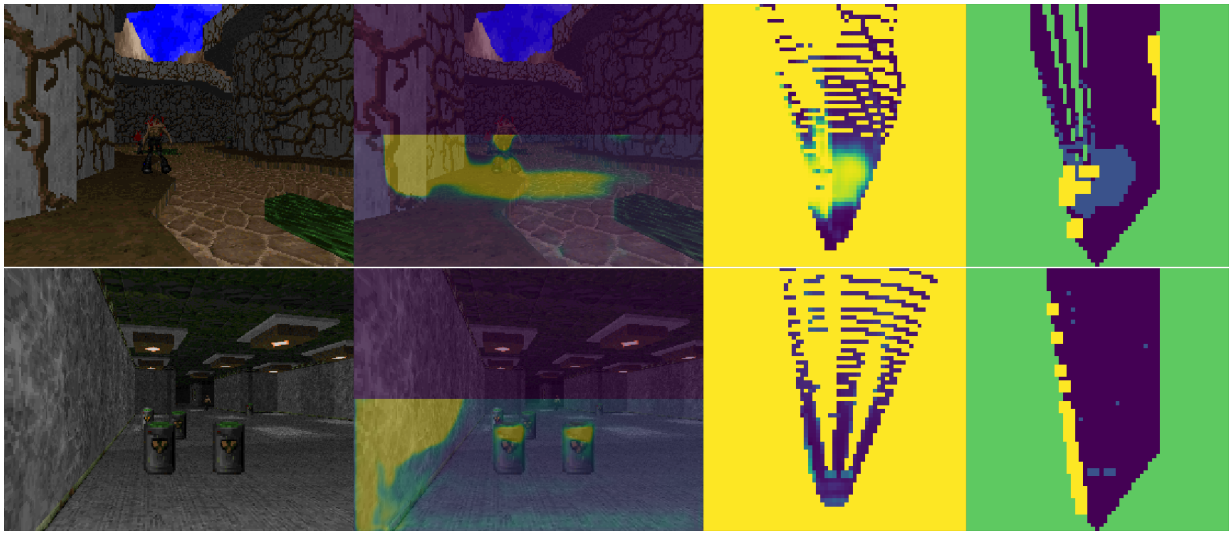

Figure 2.12: Examples of learned margins for visual signatures associated with dynamic actors. From left to right: the first image shows a RGB view of the scene, the second image shows predicted affordances $\hat{y}$ overlaid on top of the RGB view, the third image shows the projected confidence map, and the last image shows the cost map used to plan the optimal path. From the first example, it can be seen that regions that are spatially close to dynamic actors are associated with higher traversal costs in the final cost map, akin to a "margin of safety". The second example shows that static hazards/obstacles such as barrels are not associated with substantial affordance margins.

collisions during the initial stages of active training, stopping the agent and providing labels whenever an undesirable state is entered. Importantly, since data from these expensive sensors is not directly used as an input by the model, they can be removed once a satisfactory model has been trained; production-spec robots are free to employ low-cost sensing without the need for high-cost feedback sensors.

Additionally, there exist many scenarios in which feedback sensors can help label examples without the need to experience catastrophic failures such as high-speed collisions. One example is the discrepancy between wheel speed sensor values, which can be used to detect loss of traction on a wheeled robot when travelling over rough or slippery surfaces. By collecting observation-label pairs, we could then learn affordance maps to help such a robot navigate over the smoothest terrain.

Finally, we would like to emphasize that in scenarios where it is difficult to obtain oracle-labelled data and "trial and error" approaches are employed by necessity, we have shown that our proposed approach is many times more sample efficient that previous PPO-based reinforcement learning approaches for mobile robotics [17] (which also suffer from the same types of problems). If collecting a sample is costly due to the burden of operational hazards, we argue that a reduction in the number of samples required translates to an improvement in overall safety.

2.5.3 Reproducibility and Extensibility

The active affordance learning (A2L) framework, as proposed in this chapter, contains a large number of moving parts and tunable hyper-parameters that can significantly impact navigation performance. In order to facilitate future research and ensure that experiments are reproducible, we have released an open-source implementation[1] of A2L. To facilitate ease of evaluation, we also include the full VizDoom dataset, model hyper-parameters, as well as trained models.

[1] A2L implementation accessible from: https://github.com/wqi/A2L

Chapter 3

What-If Motion Prediction for Autonomous Driving

3.1 Motivation

Forecasting or predicting the future states of other actors in complex social scenes is a central challenge in the development of autonomous vehicles (AVs). This is a particularly difficult task because actor futures are multi-modal and depend on other actors, road structures, and even the AV's intended motion plan. The emergence of large-scale AV testing, together with the public release of driving datasets and maps [11, 15, 38, 68], has stimulated promising recent work on data-driven feedforward approaches [3, 14, 19, 24, 57, 70] designed to address these challenges.

Representations. Most approaches embed both social and map information within a birds-eye-view (BEV) *rasterized image*, allowing learned models (typically a combination of CNNs and RNNs) to predict trajectories from extracted features. Although convenient, there are some drawbacks to rasterization: 1) the resulting models tend to require a large number of parameters [29] and 2) some facets of the problem are best represented in coordinate spaces that are not conducive to rasterization. For example, while the physics of vehicle motion are generally modeled in Euclidean space, lane-following behaviors and map-based interactions are easier to represent in curvilinear coordinates of the road network [15]. Similarly, social interactions between N actors can be captured naturally in a topological graph representation with N nodes; notable recent methods VectorNet [29] and SAMMP [50] take such an approach, representing individual objects as nodes that may attend to one another.

Explainability. While the strong benchmark performance of feedforward models is encouraging, safety critical applications may require top-down feedback and causal explainability. For example, because the space of all potential futures in real-world urban driving settings is quite large, real-time planning may require the ability for a planner to interactively probe the forecaster, exploring only those futures that are relevant for planning (see Fig.3.1a). Approaches that require re-generation or re-processing of the scene context in order to explore alternate futures may be too inefficient for real-time planning.

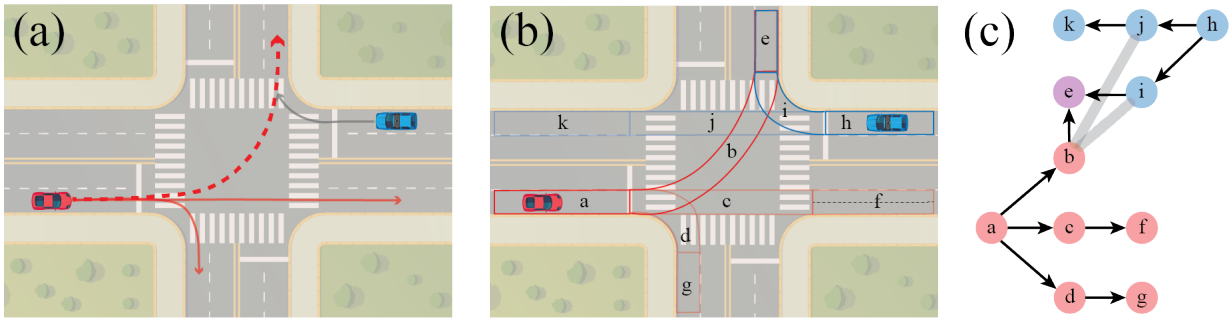

Figure 3.1: While many feasible futures may exist for a given actor, only a small subset may be relevant to the AV's planner. In **(a)**, neither of the dominant predicted modes (solid red) interact with the AV's intended trajectory (solid grey). Instead, the planner only needs to consider an illegal left turn across traffic (dashed red). **(b)** depicts a partial set of lane segments within the scene; illegal maneuvers such as following segment b can either be mapped or hallucinated. A centerline (centered polyline) associated with a lane segment is shown in segment f (dashed black). The planner can utilize the directed lane graph **(c)** to identify lanes which may interact with its intended route. Black arrows denote directed edges, while thick grey undirected edges denote conflicting lanes. Such networks are readily available in open street map APIs [33] and the recently-released Argoverse [15] dataset.

Our Approach. In this chapter, we develop a RNN-based approach for context-aware multi-modal behavior forecasting. Our approach does not require rasterized input and includes both a road-network attention module and a dynamic interaction graph to capture interpretable geometric and social relationships. In contrast to existing graph-based approaches [29, 50], we structure our model to efficiently support counterfactual reasoning. The social context of individual agents can be manipulated in order to condition upon additional hypothetical (unobserved) actors or to ablate specific social influences (Fig. 3.5). We make intimate use of the road *network*, generating topological goals in the form of lane polylines that are constructed from the underlying directed graph of lane segments (Fig. 3.1c). Importantly, rather than encoding the full local map structure, we explicitly condition forecasts upon individual topological goals. This allows the planner to reason about and query for relevant trajectories (e.g. "reforecast that actor's motion given the left turn intersecting my path"). To our knowledge, we are the first to demonstrate counterfactual forecasts based on such topological queries.

3.2 Related Work

State-of-the-art models for multi-agent motion forecasting borrow heavily from both the natural language (sequence models) and computer vision (feature learning) communities. Although an extensive body of relevant work exists, the seq2seq [69] and ResNet [34] architectures are of particular theoretical and practical importance. Our work is most related to methods that forecast from intermediate representations such as tracking output [50], although a significant body of work that operates directly on sensor input also exists [13, 47]; we do not explore such approaches in-depth.

Motion Forecasting. Until recently, motion forecasting research has primarily focused on pedestrian trajectories, either in the context of first-person activity recognition [39], sports [80], or multi-actor surveillance [59]. Social-LSTM [1] introduced social pooling within a RNN encoder-decoder architecture, providing a template to address varying numbers of actors and permutation problems caused by social input. Extensions such as *SoPhie* [61] have leveraged features extracted from the physical environment, attention mechanisms, and adversarial training. DESIRE [43] proposed a scene-context fusion layer that aggregates interactions between agents and the scene context.

Conditional Forecasting. Recent work has also investigated forecasting models conditioned on intent: [14, 55] condition the agent's forecast on a predefined set of anchor trajectories, while [21, 22] treat forecasting as a classification problem, first predicting a high level maneuver before conditioning predictions on that maneuver. Other methods, such as [70], predict a conditional probability density over the trajectories of other actors given a hypothetical rollout of the focal agent. Our work is most related to PRECOG [58], which demonstrates that conditioning on an actor's goal alters the future states of other actors within the scene (similar to polyline conditioning). Our approach, however, requires no rasterized input and can efficiently alter the social context as well.

Rasterization. Popular methods for AV forecasting [3, 24] have employed complex rasterized representations of scene context, constructing BEV images of the surrounding environment by combining trajectory histories with rich semantic information (lanes, speed limits, traffic light states, etc.) from maps. Although some of these methods [20, 24, 55] generate all predicted states simultaneously, others [3, 14, 15, 43, 50, 70] employ a recurrent decoder to predict states sequentially; [35] experiments with both approaches. More recently, there has been interest towards rasterization-free approaches for capturing scene context; [50] uses multi-head attention to encode interactions between social actors. We adopt a similar approach, attending over lane polylines from the map, in addition to the social graph.

Graph Neural Networks. Graph neural networks and graph convolution have emerged in response to problems that cannot be easily represented by matrices of pixels or simple vectors. Architectures vary widely in response to diverse underlying problems and we refer the reader to [75] and [5] for a comprehensive review. Our work is built upon graph attention networks (GATs), introduced in [73]. VectorNet [29] is a closely related method, which proposes a deep graphical model over individual road components and agent histories (represented as sets of vectors), claiming a 70% reduction in model size compared to rasterized counterparts. Our work features similar advantages in parameter efficiency, but represents lane polylines as ordered sequences of points. Additionally, VectorNet conditions over the entire local neighborhood (e.g. left turn lane, right turn lane, neighboring lanes etc.) and is consequently not structured for counterfactual reasoning over specific map elements (e.g. right turn lane). Finally, VectorNet employs a deterministic decoder limited to a single trajectory. In contrast, our approach employs a multi-modal decoder capable of generating diverse predictions.

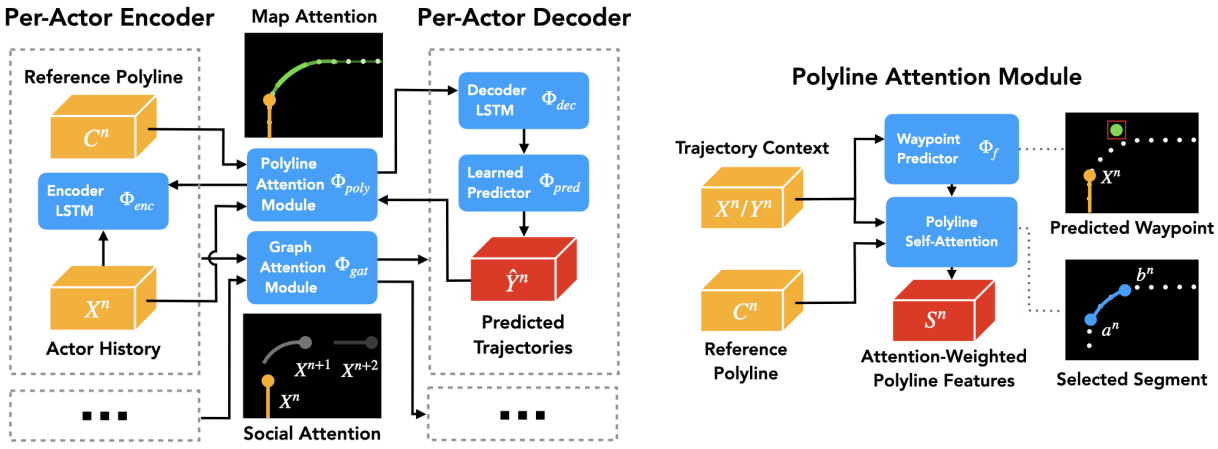

Figure 3.2: Overview of the data flow within the WIMP encoder-decoder architecture **(left)** and polyline attention module **(right)**. Input trajectories and reference polylines are first used to compute per-actor embeddings; social context is then incorporated via graph attention. Finally, a set of predictions is generated using a map-aware decoder that attends to relevant regions of the polyline via soft-attention.

3.3 Method

Our proposed architecture, the *what-if* motion predictor (WIMP), addresses the task of motion forecasting by learning a continuous-space discrete-time system with N interacting actors. Let $\mathbf{x}_t^n \in \mathbb{R}^2$ denote the n-th actor's planar (x, y) coordinates at time t and $\mathbf{X}_t \doteq \{\mathbf{x}_t^1, \mathbf{x}_t^2, \ldots, \mathbf{x}_t^N\}$ denote the joint state of all N actors. Let $\mathbf{X} \doteq \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_t\}$ denote the joint observable history up until time t and $\mathbf{X}^n = \{\mathbf{X}_1^n, \mathbf{X}_2^n, \ldots, \mathbf{X}_t^n\}$ represent the entire observable history for actor n. Analogously, let $\mathbf{Y} \doteq \{\mathbf{Y}_{t+1}, \mathbf{Y}_{t+2}, \ldots, \mathbf{Y}_{t+T}\}$ denote the joint state of all actors for future time-steps $t+1$ to $t+T$. Let $\mathbf{Y}_t$, $\mathbf{Y}^n$, and $\mathbf{y}_t^n$ be defined accordingly.

Road Network Representation via Polylines. Popular approaches for motion forecasting often rely on rasterized representations to provide contextual information about scene and road geometry [3, 24, 70]. Instead, we represent a valid path through the road network (directed graph of lane segments) using the concatenated center polylines of each road segment. Conditioning on polyline-based inputs has several advantages over its rasterized counterpart: i) it provides a strong, evidence-based prior for accurate predictions, ii) it allows for interpretable model behaviour analysis and enables counterfactual predictions that condition on hypothetical "what-if" polylines (see Section 3.4.3), and iii) it leads to more memory efficient models that do not require image-processing components.

We represent the reference polyline that guides actor n as a set of P discrete points $\mathbf{C}^n = \{\mathbf{c}_1^n, \mathbf{c}_2^n, \ldots, \mathbf{c}_P^n\}$, where $\mathbf{c}_i^n \in \mathbb{R}^2$; the collective set of such polylines for all actors is denoted by $\mathbf{C} = \{\mathbf{C}^1, \mathbf{C}^2, \ldots, \mathbf{C}^N\}$. Polyline C^n is obtained by searching the road network along the direction of motion for the highest similarity lane segment to X_n (additional details provided in Section 3.3.6). The final objective is to effectively model the conditional distribution $\Pr(\mathbf{Y}|\mathbf{X}, \mathbf{C})$; though it is possible to model the aforementioned distribution in a joint fashion, it

is often intractable and computationally inefficient for large N. Similar to [50, 70], we employ a RNN-based architecture to sequentially model $\Pr(\mathbf{Y}|\mathbf{X}, \mathbf{C})$. Specifically, we assume that the following factorization holds:

$$\Pr(\mathbf{Y}|\mathbf{X}, \mathbf{C}) = \prod_{\delta=t+1}^{t+T} \Pr(\mathbf{Y}_\delta|\mathbf{Y}_{t+1}, \ldots, \mathbf{Y}_{\delta-1}, \mathbf{X}, \mathbf{C}) = \prod_{\delta=t+1}^{t+T} \prod_{n=1}^{N} \Pr(\mathbf{y}_\delta^n|\mathbf{Y}_{t+1}, \ldots, \mathbf{Y}_{\delta-1}, \mathbf{X}, \mathbf{C}^n) \tag{3.1}$$

It should be noted that even though Eq. 3.1 factorizes as a product of conditionals over individual actors conditioned on individual polylines, global information regarding other actors and polylines is *implicitly* encapsulated via the history $\mathbf{X}$ and previous predictions $\{\mathbf{Y}_{t+1}, \ldots, \mathbf{Y}_{\delta-1}\}$. To capture this distribution, we propose a novel recurrent, graph-based, attentional approach. As shown in Fig. 3.2, the WIMP architecture has three key components: i) a graph-based encoder that captures scene context and higher-order social interactions, ii) a decoder that generates diverse, multi-modal predictions, and iii) a novel polyline attention mechanism that selects relevant regions of the road network to condition on. Next, we will describe each of these components in detail.

3.3.1 Historical Context via Recurrence

$$\mathbf{h}_t^n = \Phi_{enc}\left(\mathbf{x}_t^n, \mathbf{s}_t^n, \mathbf{h}_{t-1}^n\right) \quad , \quad \mathbf{s}_t^n = \Phi_{poly}\left(\mathbf{C}^n, \mathbf{x}_t^n, \mathbf{h}_{t-1}^n\right) \tag{3.2}$$

Each actor's contextual history $\mathbf{h}_n^t$ is captured via a shared recurrent encoder Φ_{enc}. Similar to [70], we also employ a *point-of-view* transformation $\Gamma(\mathbf{X^n})$ to normalize each actor's history to a reference frame by translation and rotation such that the $+x$-axis aligns with a focal agent F's heading (such as the AV) and $\mathbf{x}_1^F = (0, 0)$.

3.3.2 Geometric Context via Polyline Attention

As described in Eq. 3.2, each actor n attends to segments of their reference polyline $\mathbf{C}^n$ through the learned function Φ_{poly}. Intuitively, drivers pay attention to areas of the road network that they are *currently* close to, as well as future *goal* locations that they plan to reach. Φ_{poly} operationalizes this intuition by predicting, for each actor n and timestep t, a current and goal index along its polyline:

$$a_t^n = \arg\min_p \left\{d\left(\mathbf{c}_p^n, \mathbf{x}_t^n\right)\right\} \quad , \quad b_t^n = \arg\min_p \left\{d\left(\mathbf{c}_p^n, \Phi_f\left(\mathbf{x}_t^n, \mathbf{h}_{t-1}^n, \Delta\right)\right)\right\} \tag{3.3}$$

where $d(\cdot)$ is a distance metric and Φ_f is a learned function that hallucinates a coarse *waypoint* Δ time-steps in the future. It should be noted that Φ_f doesn't make use of any polyline information and predicts the waypoint solely based on kinematic history; training is conducted in a self-supervised manner using ground-truth future trajectories as labels. The vectorized attention-weighted representation $\mathbf{s}_t^n$ for the segment $\bar{\mathbf{C}}_t^n$ between current and goal indices can then be

obtained as follows (where $\mathbf{Q}, \mathbf{V}, \mathbf{K}$ are learned transformation matrices, similar to those employed in [50]):

$$\Phi_{poly}(\mathbf{C}^n, \mathbf{x}_t^n, \mathbf{h}_{t-1}^n) = \sum_{r \in [a_t^n, b_t^n]} \upsilon_{tr}^n \mathbf{V}\mathbf{c}_r^n \quad , \quad \upsilon_{tr}^n = \underset{r}{\text{softmax}} \left(\mathbf{Q}\mathbf{h}_{t-1}^n \odot \mathbf{K}\mathbf{c}_r^n \right) \tag{3.4}$$

3.3.3 Social Context via Graph Attention

As Φ_{enc} runs independently over all actors, the hidden representation obtained after t time-steps $\mathbf{h}_t^n$ for a particular actor n is oblivious to other dynamic participants in the scene. One possible solution is to provide $\mathbf{x}_t^i; \forall i \neq n$ as an input to Eq. 3.2, but this is computationally inefficient and memory intensive. Instead of capturing social interactions in the planar coordinate space, we leverage the ability of Φ_{enc} to generate rich latent hidden representations $\mathbf{h}_t^n$ for a particular actor n. Inspired by [73], we employ a graph attention module Φ_{gat} that operates over these representations as follows:

$$\bar{\mathbf{h}}_t^n = \sigma \left(\mathbf{h}_t^n + \frac{1}{D} \sum_{d=1}^{D} \sum_{j \in N \setminus n} \alpha_{nj}^d \mathbf{W}^d \mathbf{h}_t^j \right) \ , \ \alpha_{nj}^d = \underset{j}{\text{softmax}} \left(\mathbf{a}^d \odot \left[\mathbf{W}^d \mathbf{h}_t^n, \mathbf{W}^d \mathbf{h}_t^j \right] \right) \tag{3.5}$$

where D is a hyperparameter denoting the number of attention heads, $[\cdot, \cdot]$ is the concatenation operation, $\odot$ is the inner product, and $\mathbf{W}^d$, $\mathbf{a}^d$ are learned parameters. Note that there is a subtle difference between Eq. 3.5 and the architecture proposed in [73], wherein, for each agent n, we focus on learning a residual change to its *socially-unaware* hidden representation $\mathbf{h}_t^n$. Intuitively, this can be thought of as an actor initially having a socially-agnostic estimate of its future trajectory, with Φ_{enc} learning a residual change to incorporate information from other actors within the scene.

3.3.4 Decoding

Following Eq. 3.1, WIMP aims to learn the conditional distribution $\Pr(\mathbf{y}_\delta^n | \mathbf{Y}_{t+1}, \ldots, \mathbf{Y}_{\delta-1}, \mathbf{X}, \mathbf{C}^n)$ for each actor n. To achieve this goal, we employ a LSTM-based decoder Φ_{dec} that: i) generates diverse and multi-modal predictions, and ii) conditions each prediction on a reference polyline $\mathbf{C}^n$. Particularly, for a future time-step δ, we can obtain $\mathbf{y}_\delta^n$ as follows:

$$\mathbf{y}_{\delta+1}^n = \Phi_{pred} \left(\mathbf{o}_\delta^n \right) \quad , \quad \mathbf{o}_\delta^n, \bar{\mathbf{h}}_\delta^n = \Phi_{dec} \left(\mathbf{Y}_\delta, \bar{\mathbf{s}}_\delta^n, \bar{\mathbf{h}}_{\delta-1}^n \right) \quad , \quad \bar{\mathbf{s}}_\delta^n = \Phi_{poly} \left(\mathbf{C}^n, \mathbf{y}_\delta^n, \bar{\mathbf{h}}_{\delta-1}^n \right) \tag{3.6}$$

where Φ_{pred} is a learned prediction function and Φ_{poly} is a polyline-attention module as described in Section 3.3.2. We note that the implementation of Φ_{pred} is architecturally agnostic; for example, Φ_{pred} could be a bivariate Gaussian as in [70], or a mixture of Gaussians as in [50]. For datasets like Argoverse [15] that only evaluate predictions for a single focal actor F, decoder input $\mathbf{Y}_\delta$ might only contain predictions for a single actor $\mathbf{y}_\delta^F$. However, even in this scenario, WIMP is still able to model social interactions via embeddings $\bar{\mathbf{h}}_t^n$ obtained from the graph-based encoder.

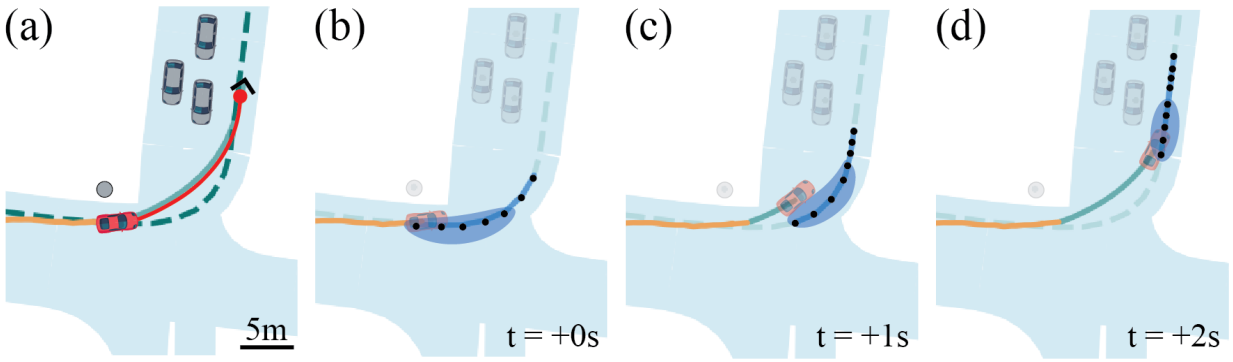

Figure 3.3: Visualizing the *map lane polyline* attention weights generated during decoding. In the scenario depicted in **(a)**, the focal actor's history is shown in yellow and its ground-truth future in red. The red circle highlights the true state 3s into the future. The solid green line denotes a predicted trajectory with a black chevron marking the $t = +3s$ state. The dashed green line shows the reference polyline. Grey cars/circles illustrate the current positions of on/off roadway actors. In **(b, c, d)**, opacity corresponds to the magnitude of social attention. The subset of the polyline selected by the polyline attention module is shown in solid blue, and the attention weights on points (black circles) within that segment are shown via an ellipse (for predictions at $t = +0s, +1s, +2s$ respectively). WIMP learns to attend smoothly to upcoming points along the reference polyline.

3.3.5 Learning

WIMP is trained on collections of triplets containing: historical trajectories, ground-truth future trajectories, and map-based road context $\{(\mathbf{X}, \mathbf{Y}, \mathbf{C})\}$. Following standard forecasting benchmarks, we only predict the future trajectory for a single *focal* agent in each training example, denoted as $\mathbf{Y}^F$.

Winner-Takes-All: To encourage diversity and multi-modality in the set of predicted trajectories, we learn a mixture of M different predictors. Diversity is encouraged through a "multiple choice" [31] or "winner-takes-all" loss that explicitly *assigns* each training example to a particular mixture:

$$\text{loss} = \min_{m \in \{1 \ldots M\}} ||\hat{\mathbf{Y}}_m^F - \mathbf{Y}^F|| \tag{3.7}$$

where $\hat{\mathbf{Y}}_m^F$ is the focal trajectory predicted by the m^{th} mixture. Having experimented with various distance functions, we found the $L1$ norm between trajectories to perform well. We also experimented with *multi-agent* prediction of future trajectories for *all* actors, but did not observe improved performance. We posit that this may be due to the large numbers of parked actors present in urban driving scenarios, which may require different representations or larger capacity forecasting models.

Optimization: By keeping track of the $\arg\min\, m$ index for each training example, WTA loss naturally clusters training examples into M sets. Previous work has shown that directly optimizing this loss can lead to poor results because (a) it is difficult to optimize stochastically with

mini-batch SGD, as the optimization is sensitive to initialization and (b) each mixture can be prone to overfitting, as it is trained with less data. One proposed solution is “evolving WTA" (EWTA) [48], where the single minimum $\min_m$ is replaced with the M' lowest-cost mixtures. Initializing with $M' = M$, examples are initially associated with all M clusters, encouraging every mixture to generate identical predictions. Over time, as M' is annealed to 1 (resulting in standard WTA loss), iterative specialization of each mixture ensures that each of the final mixtures has been “pre-trained" with the full dataset.

Mixture Ranking: The above produces M different predicted trajectories, which can be fed directly into multi-output forecasting benchmarks that require methods to return M predictions. To repurpose these outputs for single-prediction evaluations, we rank each mixture's accuracy on a validation set.

3.3.6 Polyline Selection

To obtain relevant reference trajectories from the underlying vector map, we employ a heuristic-based polyline proposal module based on code released in the Argoverse API [15]. Using either the observed (0-2s) history of the focal actor (during evaluation) or the full (0-5s) ground truth trajectory (during training), we query the proposal module for a ranked list of candidate polylines sorted by similarity to the reference trajectory. These candidate polylines are obtained through the following procedure:

1. **Find Candidate Lanes**: We first search the map lane graph to find the set of all lanes containing nodes that are located within a 2.5m distance from the last point of the query trajectory. If no lanes are found, we iteratively expand the search radius by a factor of 2 until at least one candidate lane is identified.

2. **Construct Candidate Polylines**: For each candidate lane node returned in the above set, we construct corresponding polylines by recursively traversing the lane graph through successor and predecessor nodes, stopping once a distance threshold has been reached in both directions. In our implementation, we set this distance threshold to be $2\times$ the total length of the query trajectory. We then connect the traversed nodes with directed edges (from earliest predecessor to latest successor), forming a polyline composed of individual points. To show how candidate polylines L_c are constructed from candidate lane node A:

   Enumerated Successors: *{(A->B->C), (A->D->E)}*
   Enumerated Predecessors: *{(F->G->A), (H->I->A)}*
   Constructed Lane Polylines:
   L1 : F->G->A->B->C
   L2 : H->I->A->B->C
   L3 : F->G->A->D->E
   L4 : H->I->A->D->E
   L_c*: {L1, L2, L3, L4}*

3. **Remove Overlapping Polylines**: Next, we filter the set of candidate polylines constructed in the previous step by removing polylines that overlap significantly with other candidates.

4. **Sort By Point-in-Polygon Score**: We then sort the filtered set of candidate polylines by point-in-polyline (PIP) score, defined as the number of query trajectory points that lie within the polygon formed by lane regions corresponding to each polyline. If there are n points in the query trajectory, the PIP score is bounded to the range $[0, n]$. To give a concrete example, if $n = 20$ and PIP scores for candidate polylines L_c are *{L1: 15, L2: 10, L3: 5, L4: 20}*, the sorted list of candidate polylines will be returned in the order L_{pip} = *[L4, L1, L2, L3]*.

5. **Sort By Polyline-Trajectory Alignment**: We also sort the filtered set of candidate lines by a polyline-trajectory alignment-based score. To compute this score, the query trajectory is first mapped to the 2D polyline-based curvilinear coordinate system (as defined in Argoverse), wherein axes are defined to be tangential and perpendicular to a reference polyline. We define the alignment score to be the maximum tangential distance reached along the query trajectory (better alignment results in longer distances travelled along the reference polyline). To give a concrete example, if the maximum tangential distance for each of the candidate polylines is *{L1: 10 , L2: 25, L3: 2, L4: 20}*, the sorted list of candidate polylines will be returned in the order L_a = *[L2, L4, L1, L3]*.

6. **Selecting Polylines**: We sort candidate polylines using two different methods of scoring because examining PIP score in isolation can sometimes be misleading. For example, a car moving slowly across an intersection could result in high PIP scores being assigned to polylines obtained from lanes with orthogonal directions of travel. Using the polyline-trajectory alignment score alone can also result in similar confusion. For example, a nearby protected turn lane that is parallel to the query trajectory's direction of travel may be assigned a high alignment score, even if the polyline represents a semantically different future.

   For this reason, it is important to rank polyline proposals using a combination of both metrics. In our implementation, we employ a heuristic-based selection process, wherein the polylines are drawn from the top of L_{pip} and L_a in alternating order. To give a concrete example, in the scenario we have posed above, querying for the best 2 polylines would return L = *[L4, L2]* (where *L4* is the best PIP polyline and *L2* is the best-aligned polyline).

7. **Using Proposed Polylines**: During training, we query for and use only the top-ranked polyline proposal from each set. However, at inference time, it is possible to trade off prediction diversity and accuracy by controlling the the number of polyline proposals and predictions generated per polyline (e.g. 6 predictions conditioned on 1 polyline vs. 1 prediction conditioned on each of 6 polylines).

3.4 Experiments

We demonstrate the effectiveness of WIMP at generating accurate, interpretable, and controllable trajectory predictions for roadway actors. We first show that the scene attention encoder is capable of capturing the complex contextual, semantic, and social relationships that are present in real-world urban driving scenarios. These learned scene embeddings can be combined with

multi-modal decoders to generate a diverse set of plausible future trajectories. We then perform a series of counterfactual reasoning-based experiments to demonstrate how the distribution of predicted modes is influenced by scene context. The implementation details and hyperparameters are provided in Section 3.5.2.

3.4.1 Experimental Setup

Datasets. We conduct our experiments using the Argoverse [15] motion forecasting dataset, a large scale vehicle trajectory dataset containing more than 300,000 curated scenarios extracted from vehicle logs in urban driving scenarios. Given a 2 second trajectory history as input, the goal is to predict the future motion of a particular focal agent over the next 3 seconds (sampled at $\approx$ 100 ms intervals). In addition to the focal agent history, location histories of nearby (social) actors are also provided. Importantly, Argoverse includes a semantic vector map composed of lane-based polylines.

Although the Argoverse dataset provides a high volume of *interesting* data for both training and evaluation, the focal trajectories are not particularly diverse in terms of directional variation, with more than 80% of scenarios featuring straight line trajectories over the full 5 second window. In order to evaluate how WIMP performs in the presence of uncertainty, we also extract a small subset ($\approx$350 examples) of particularly challenging scenarios that are characterized by blind turns (defined as examples where the observed 2-sec. trajectory is straight, but the ground truth future 3-sec. trajectory contains a turn and/or lane change). Even for recent state-of-the-art methods, the blind turn (BT) subset presents a significant challenge, as generation of high-quality predictions necessitates the incorporation of both social and semantic information to resolve uncertainty.

In addition to Argoverse, we also evaluate using the NuScenes prediction dataset, which contains a similar collection of approximately 40,000 scenarios that were extracted from 1,000 curated scenes. These scenarios were collected within two distinct regions on different continents, featuring trajectories from both left-hand and right-hand drive locales. Due to the geographic diversity of data and more significant representation of complex scenarios such as turns and intersections, NuScenes presents a more challenging prediction task than the base Argoverse dataset.

Metrics. To evaluate prediction quality, we make use of widely adopted forecasting metrics: minimum average displacement error (ADE) and minimum final displacement error (FDE) [15], evaluated for both single ($K = 1$) and multi-modal ($K = 6$) prediction scenarios. To capture prediction performance in more challenging scenarios, we also adopt the miss rate (MR) metric: the fraction of scenarios with FDE $> 2m$.

3.4.2 Argoverse Motion Forecasting

Quantitative Results. We compare WIMP to several recent state-of-the art (SOTA) methods: SAMMP [50] (best self-attention-based model, joint-winner of the 2019 Argoverse Forecasting Challenge), UULM-MRM (best rasterization-based model, joint-winner of the 2019 Argoverse Forecasting Challenge), and VectorNet [29] (a recent polyline-based model). Evaluating on the

Argoverse challenge test set (results summarized in Table 3.1), we show that each of these methods is highly competitive, performing far above the bar set by K-NN and LSTM based baselines. We further show that WIMP-based models are able to out-perform all existing published methods in both single and multi-modal prediction-based metrics.

MODEL	MR(K=6)	FDE(K=6)	ADE(K=6)	FDE(K=1)	ADE(K=1)
SAMMP [50]	0.19	1.55	0.95	4.08	1.81
UULM-MRM	0.22	1.55	0.96	4.32	1.97
NN + MAP(PRUNE) [15]	0.52	3.19	1.68	7.62	3.38
LSTM + MAP(PRIOR) [15]	0.67	4.19	2.08	6.45	2.92
VECTORNET[29]	-	-	-	4.01	1.81
WIMP ($M = 1$)	-	-	-	**3.89**	**1.78**
WIMP ($M = 6$)	**0.17**	**1.42**	**0.90**	4.03	1.82

Table 3.1: Motion forecasting performance evaluated on the Argoverse test set, with MR and minimum FDE/ADE reported for both single ($K = 1$) and multi-modal ($K = 6$) prediction scenarios.

Evaluation in Challenging Scenarios. As the overall Argoverse dataset is biased towards simple straight line trajectories, we also evaluate prediction performance on the BT subset (results summarized in Table 3.2), which consists primarily of challenging blind turn scenarios. In this setting, we show that WIMP out-performs non-map-based approaches (such as SAMMP) by a much larger margin than across the full dataset, as polyline and social graph-based attention allows the model to resolve and account for uncertainty even in complex scenarios with multiple feasible future trajectories. In such scenarios, models employing polyline-based coordinate systems, such as LSTM + Map (Prior) from [15]), also perform surprisingly well, as the prediction space is strongly conditioned on map information, trading overall performance for better turn prediction results. We note that WIMP is significantly less impacted by this bias-variance trade-off, delivering top performance in both BT and general settings. We also demonstrate that prediction accuracy improves with reference polyline quality. By employing an oracle to select the optimal polyline in hindsight (after observing the future), we observe significant improvements, indicating that WIMP can take advantage of "what-if" polylines provided by such oracles. We analyze this further in the next section.

Ablation Study In order to demonstrate how each component of the WIMP architecture contributes to overall prediction performance, we perform an ablation study and summarize the results in Table 3.3. We obtain best results when the model is provided with both map and social context, while coupled to a L1-based EWTA loss [48]. We also experiment with alternative loss formulations: replacing EWTA loss with negative log likelihood (NLL) significantly degrades performance, while standard L1 loss provides impressive ($K = 1$) performance but cannot be adapted to make multiple predictions.

NuScenes Motion Forecasting. To evaluate the generalizability of our proposed prediction architecture, we also compare WIMP to several recent learning-based methods and a physics-based

MODEL	MR(K=6)	FDE(K=6)	ADE(K=6)
SAMMP	0.67	4.91	2.38
NN + MAP (PRUNE)	0.61	5.11	3.93
LSTM + MAP (PRIOR)	0.51	2.64	3.01
WIMP	0.49	3.52	1.62
WIMP (ORACLE)	**0.33**	**2.46**	**1.30**

Table 3.2: Motion forecasting performance evaluated on the Argoverse BT validation set. As the selected data is inherently multi-modal, we only report metrics for ($K = 6$) predictions. SAMMP results were obtained from our implementation of [50], using hyper-parameters shared with WIMP.

CONTEXT	LOSS	MR(K=6)	FDE(K=6)	ADE(K=6)	FDE(K=1)	ADE(K=1)
MAP + SOCIAL	EWTA	**0.12**	**1.14**	**0.75**	3.19	1.45
MAP + SOCIAL	L1	-	-	-	**3.01**	**1.40**
MAP + SOCIAL	NLL	0.23	1.61	1.07	6.37	1.41
SOCIAL	EWTA	0.16	1.39	0.86	5.05	1.61
MAP	EWTA	0.16	1.38	0.85	3.80	1.69
NONE	EWTA	0.23	1.70	0.95	5.86	1.87

Table 3.3: Ablation studies for WIMP with different input configurations and training objectives. Quantitative results reported for $K = 1$ and $K = 6$ metrics on the Argoverse validation set.

baseline on the NuScenes prediction dataset; results are summarized in Table 3.4. Without any hyper-parameter tuning or changes to model architecture (compared to the model evaluated on Argoverse), WIMP achieves state-of-the-art results in both single ($K = 1$) and multi-modal ($K = 5, 10$) prediction scenarios, out-performing all previous methods in both miss rate and displacement error. WIMP delivers especially strong results on NuScenes due to the high proportion of intersection and turn-based scenarios, which are difficult to solve without integration of both map and social context.

3.4.3 Counterfactual Validation

Our proposed approach to conditional forecasting readily supports investigations of hypothetical or unlikely scenarios (counterfactuals). This capability can be readily used by a planner to allocate computation to only relevant futures, or to reason about social influences from occluded regions of the road network. Importantly, these counterfactual queries can also be used to investigate and evaluate models beyond distance-based metrics. Sensible predictions conditioned on extreme contextual input indicates that our model has learned a powerful causal representation of driving behavior and is likely to generalize well (see Figs. 3.4 and 3.5).

MODEL	MR(K=10)	ADE(K=10)	MR(K=5)	ADE(K=5)	FDE(K=1)
LISA	0.46	1.24	0.59	1.81	8.57
TRAJECTRON++[?]	0.57	1.51	0.70	1.88	9.52
CXX	0.60	1.29	0.69	**1.63**	8.86
COVERNET[55]	0.64	1.92	0.76	2.62	11.36
PHYSICS ORACLE[55]	0.88	3.70	0.88	3.70	9.09
WIMP	**0.43**	**1.11**	**0.55**	1.84	**8.49**

Table 3.4: Motion forecasting performance evaluated on the NuScenes validation set, with MR and minimum FDE/ADE reported for ($K = 1$), ($K = 5$), and ($K = 10$) prediction scenarios. Metrics are computed using the reference implementation provided with the NuScenes devkit.

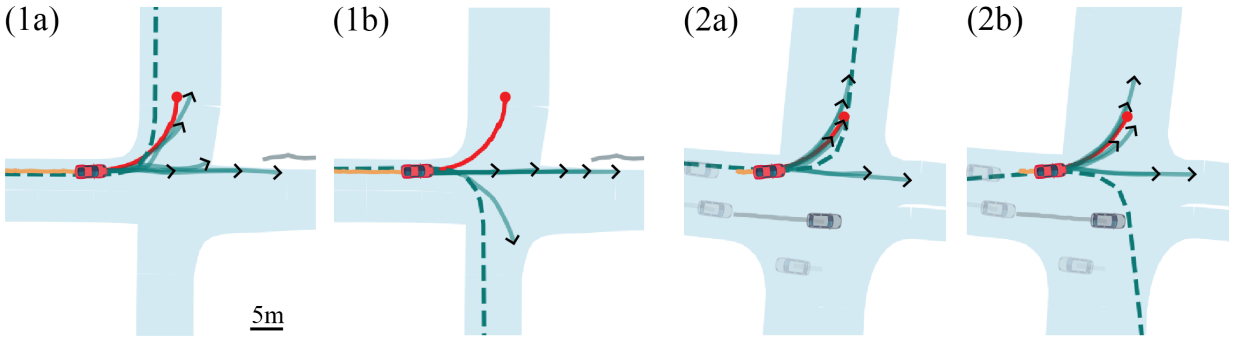

Figure 3.4: Visualizations of two prediction scenarios that condition on **(a)** heuristically-selected polylines (see Section 3.3.6 for details) and corresponding **(b)** counterfactual reference polylines. When making diverse predictions, WIMP learns to generate some trajectories independent of the conditioning polyline (see the straight through predictions in **(a)**). Additionally, if the reference polyline is semantically or geometrically incompatible with the observed scene history (as in **(2b)** where the counterfactual polyline intersects other actors), the model learns to ignore the map input, relying only on social and historical context. Visualization style follows Fig. 3.3.

3.4.4 Qualitative Results

As trajectory prediction is a fundamentally three-dimensional task that requires integration of information across space and time, it can be difficult to capture temporal context using static 2D images alone. To address this issue, we provide dynamic visualizations [1] (following the style of Fig. 3) for each of the BEV scenarios shown in the main text (Figs. 3-5).

We also include additional dynamic visualizations from prediction scenarios that capture a broad range of interesting events: acceleration, braking, full stops, fast driving, exiting driveways, lane changes, left turns, right turns, wide turns, and use of protected turn lanes. These examples are intended to demonstrate that WIMP not only generates predictions that are accurate and diverse, but also generalizes to a wide variety of geographic and semantic settings. Finally, we include a short explanatory video that demonstrates multi-modal prediction in a right turn

[1] WIMP dynamic visualizations accessible from: https://bit.ly/WIMP_dynamic

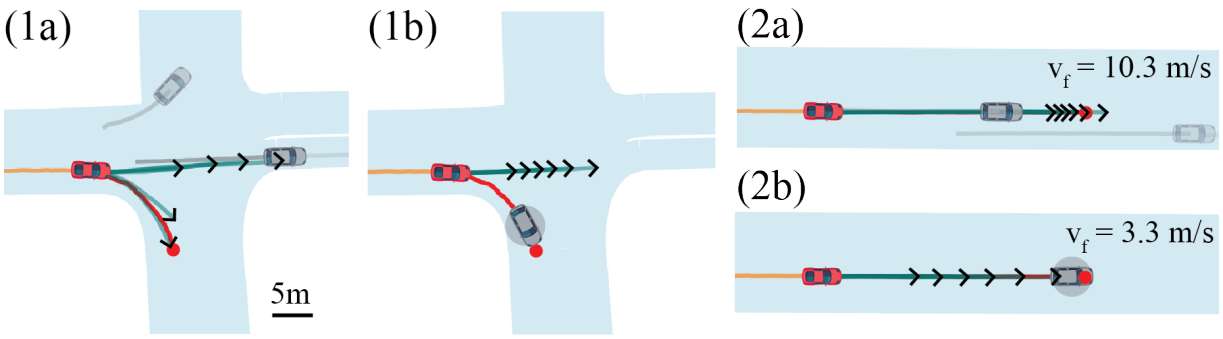

Figure 3.5: Visualizations of two scenarios that condition on **(a)** ground-truth scene context and **(b)** counterfactual social contexts (best viewed with magnification). Counterfactual actors are highlighted with a grey circle. In **(1b)**, we inject a stopped vehicle just beyond the intersection, blocking the ground-truth right turn. Given the focal agent's history and velocity, this makes a right turn extremely unlikely, and that mode vanishes. In **(2b)** we replace the the leading actor in **(2a)** with a stopped vehicle. As expected, this causes the model to predict trajectories containing aggressive deceleration. The final velocity (v_f) of a representative trajectory is 3.3m/s in the counterfactual setting, compared with 10.3m/s in the original scene. Visualization style follows Fig. 3.3.

scenario, showing how each of the components in WIMP contributes to the final output.

3.5 Discussion

In this chapter, we proposed a recurrent graph-based attentional framework with interpretable geometric and social relationships that supports injection of counterfactual contextual states. Leveraging information from historical, social, and geometric sources, WIMP facilitates joint multi-modal prediction of future states over an arbitrary number of actors within a scene, outperforming all previous methods on the Argoverse forecasting dataset. In the remainder of this section, we 1) discuss how WIMP predictions can be used as part of a feedback loop to update and improve maps and 2) explain the specific implementation details used within our experiments.

3.5.1 Improving Maps via Prediction

Planning and forecasting in AVs are tightly coupled to semantic map data that is collected, processed, and annotated offline. By examining for repeated and significant disagreement between heuristically chosen lane polylines and accurate forecasted trajectories, we can automatically identify map locations where the proposed lane polylines fail to capture the dominant modes of traffic behavior (shown in Fig. 3.6).

This ancilary benefit of explicit path-conditioning could serve as an important feedback mechanism for generation and maintenance of safe and current maps. Without updates, maps can quickly become outdated in urban environments, as active construction and development modifies the road network and induces change in traffic patterns. One such way that map updates

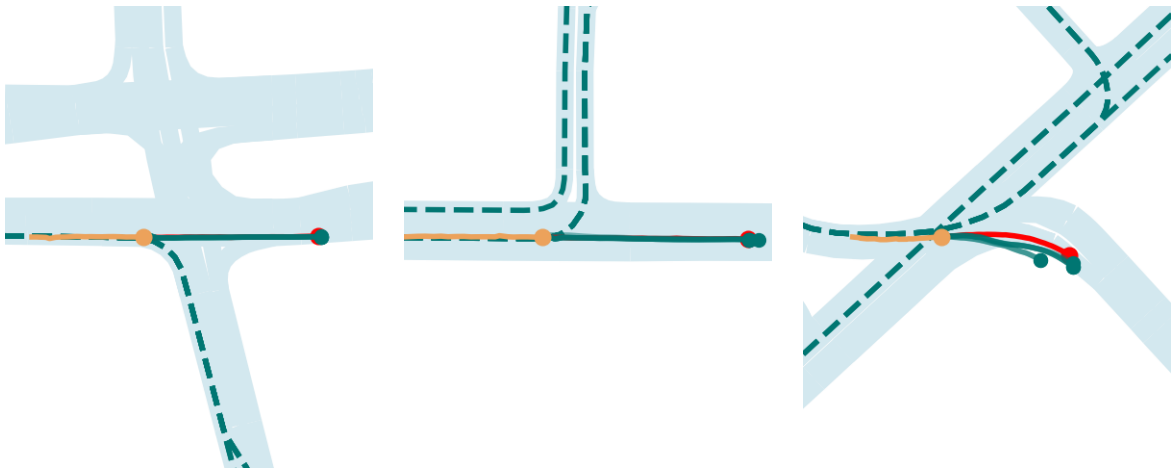

Figure 3.6: BEV visualizations of three different intersections where accurate predictions based on polyline proposals from the vector map disagree with the observed mode of traffic behavior. Visualization style follows Fig. 3.

could be performed in an online setting is to assign a weighted prior for each map polyline, with value inversely proportional to the rate of disagreement between conditioned predictions and the corresponding polyline. These polyline weights can then be used as an input to a heuristic or learning-based polyline proposal module to enable dynamic selection of high-quality reference trajectories. As variables within the environment change (e.g. construction, weather, potholes), priors can be automatically updated to capture the updated distribution of driver behavior.

3.5.2 Reproducibility and Extensibility

Although we demonstrate the WIMP forecasting framework using a vehicle trajectory prediction task in an autonomous driving setting, the architecture is designed such that concrete implementations of learned components and transformations are abstracted away. This improves generalization, as a variety of prediction tasks can be supported with just a few tweaks to model configuration. To improve reproducibility, we share the specific implementation details (including hyper-parameters for training and model configuration) for the model used in our experiments.

Normalization. Prior to all other operations, every collection of points specified in global Argoverse world coordinates (input trajectories, reference polylines, etc.) within each scenario is first transformed to a local coordinate space that is normalized with respect to focal agent F. This is implemented using an affine transformation A, such that the positive X-axis becomes aligned with the focal agent's heading (defined as the angle between $\mathbf{x}_1^F$ and $\mathbf{x}_{20}^F$) and $\mathbf{x}_1^F = (0, 0)$

Polyline Attention. Both the encoder Φ_{enc} and decoder Φ_{dec} make use of polyline attention module Φ_{poly} to capture priors provided by the map. However, weights are not shared between the two polyline attention modules. This is largely a consequence of Φ_{dec} only predicting a trajectory for the focal agent F (owing to the task formulated by the Argoverse [15] dataset), whereas Φ_{enc} takes observed trajectories from all actors as input. Φ_{poly} is implemented as a 4-layer LSTM, where the hidden state is a 4×512 dimensional vector. The distance metric $d(\cdot)$ in

Eq. 3.4 is the L2-norm. The transformations $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ defined in Eq. 3.3 are learned matrices of size 512×512 and are used in the same manner as [50, 72]. We use a dropout [67] rate of 0.5 during training, which is applied over the first three layers.

Encoder. The shared recurrent encoder Φ_{enc} used to capture each actor's location history is implemented as a 4-layer LSTM with a 512-dimensional hidden state. We use a dropout rate of 0.5 during training, which is applied over the first three layers.

Graph Attention. The graph attention module Φ_{gat} takes as input the final hidden state $\mathbf{h}_n^t$, for each actor n. Following Eq. 3.5, we set the number of attention heads D to 4, and the learned parameters $\mathbf{W}^d$ and $\mathbf{a}^d$ are of sizes 2048×512 and 1024×1 respectively.

Decoder. The decoder Φ_{dec} is configured identically to the encoder Φ_{enc}, wherein we use a 4-layer LSTM with a 512-dimensional hidden state. Following Eq. 3.6, Φ_{pred} is a linear layer that transforms the 512-dimensional output $\mathbf{o}_\delta^n$ of Φ_{dec} into a 2-dimensional prediction.

Training. For training on the Argoverse dataset, we use the ADAM optimizer with stochastic mini-batches containing 100 scenarios each; no weight decay is employed, but gradients are clipped to a maximum magnitude of 1.0. The learning rate is initialized to a value of 0.0001 and annealed by a factor of 2 every 30 epochs. We couple the optimizer to an EWTA-based loss (as described in Section 3.5), the value of M' is intialized to 6 and annealed by 1 every 10 epochs until $M' = 1$ at epoch 50. Validation metrics are computed after every 3 training epochs and training is terminated once validation metrics have failed to improve for 30 epochs in a row. Each model requires approximately 100 epochs to train on average, taking about 28 hours of compute time on an AWS "p3.8xlarge" instance equipped with 4x V100 GPUs.

Evaluation. As predictions are generated in the normalized local coordinate space, they are first transformed back to the global world space using an inverse affine transformation A^{-1} before evaluation. The minimum final displacement error (minFDE) metric is computed by taking the minimum of L2 distances between the end points of each of the k predicted trajectories and the ground truth future; minimum average displacement error (minADE) is then obtained by computing the average L2 distance corresponding to the predicted trajectory with lowest end point error. Finally, we compute the miss rate, which measures the proportion of scenarios where minFDE exceeds a threshold value (set at 2m in the Argoverse Forecasting Challenge).

Code. In order to facilitate future research and ensure that experiments are reproducible, we have released an open-source implementation[2] of WIMP, along with data processing utilities, all hyper-parameters, and trained models.

[2]WIMP implementation accessible from: https://github.com/wqi/WIMP

Chapter 4

Conclusion

4.1 Contributions

In this thesis, we addressed some of the key challenges faced by mobile robots operating in real-world, multi-actor, shared environments such as homes and roads. Focusing on motion planning and motion prediction - two important and closely-related components of the mobile robot autonomy stack, we exploited structured inductive biases to design frameworks for scene representation learning. Specifically, we leveraged large, self-supervised corpora of data to learn representations that enable goal-directed navigation within novel, unmapped environments and multi-modal vehicle motion forecasting in autonomous driving settings. Evaluating in simulation and on large-scale real-world datasets, we empirically show that use of such representations advances the state-of-the-art in these tasks, providing several possible avenues towards the deployment of safer autonomous agents.

In Chapter 2, we described a learnable approach for exploration and navigation within novel environments. Like RL-based policies, our approach learns to exploit semantic, dynamic, and even behavioural properties of the novel environment when navigating (which are difficult to capture using geometry alone). But unlike traditional RL, our approach is made sample-efficient and interpretable by way of a spatial affordance map, a novel representation that is interactively-trained so as to be useful for navigation with off-the-shelf planners. Evaluating in simulation, we show that such representations enable autonomous agents to move within hazardous, dynamic environments with significantly improved safety and efficiency over existing classical and learning-based methods.

In Chapter 3, we proposed a recurrent graph-based attentional forecasting framework with interpretable geometric and social relationships. Leveraging information from historical, social, and geometric sources, WIMP facilitates joint multi-modal prediction of future states over an arbitrary number of actors within a scene. By supporting the injection of counterfactual contextual states, we enable motion planners to pose *what-if* questions about actor behavior using topological queries such as map-based goals and social contexts. Evaluating on the large-scale Argoverse vehicle trajectory dataset, we show that WIMP provides provides significant improve-

ments in prediction accuracy, while offering unique advantages for motion planning.

4.2 Future Work

In the A2L system, we choose to encode geometric, dynamic, and semantic information relevant to motion planning within a single spatial *affordance*-based representation. Though conceptually simple, we believe that affordance maps open up further avenues for research and could help close the gap between human and autonomous navigation performance. For example, the dynamics of moving obstacles are currently captured only in an implicit fashion. A natural extension is to make this explicit, either in the form of a dynamic map or navigability module that makes use of spatio-temporal input for better affordance prediction. Another possible extension could be to combine back-projection with image-based tracking to improve the accuracy and precision of self-supervised labelling (as discussed in Section 2.5.1).

In the WIMP framework, we generate per-actor candidate polylines from the road network using similarity-based heuristics that match against the previously observed trajectory. However, such a selection process has the potential to fail in scenarios such as lane changes and merges, where an actor may transition between road lanes. In future work, this issue could be addressed by extending the polyline selection procedure with an end-to-end trainable solution that is less brittle, enabling the model to automatically select candidate polylines based on observed scene context. This procedure could potentially operate over the lane graph directly, learning when nodes in different lanes should be joined to form a transitional polyline.

Alternative directions for future research could explore applications of WIMP beyond autonomous driving, perhaps for prediction of pedestrian trajectories or human actions. The recent release of the Forking Paths dataset [45] presents particularly exciting opportunities for such research, as it provides the first publicly-released collection of true multi-future prediction scenarios. Using a realistic 3D simulator, the authors re-create scenarios based on real-world trajectory data and ask human annotators to control pedestrian actors with different latent goals. Employing a prediction architecture that explicitly conditions on semantic goals, such as WIMP, could be of particular value in such an environment.

Another exciting direction for future work exists at the intersection of affordance learning and trajectory prediction. By replacing the static affordance maps employed in A2L with a probabilistic, spatio-temporal representation that encodes a *best-guess* prediction of future navigability, it is possible to create an interpretable representation that explicitly captures the multi-modality of future states. Coupling such a representation with classical planning could enable autonomous agents to reach their goals more efficiently in the presence of nearby dynamic actors, while respecting social norms and maintaining appropriate margins of safety. Another avenue of research could be to explore actor or object-based representations for such information, which could be employed with WIMP-like encoders to capture scene context for jointly learned planners.

Bibliography

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016. 3.2

[2] Andrew Apted. Oblige level maker. `http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/`, 2017. URL `http://oblige.sourceforge.net/`. 2.4.1

[3] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018. 3.1, 3.2, 3.3

[4] Somil Bansal, Varun Tolani, Saurabh Gupta, Jitendra Malik, and Claire Tomlin. Combining optimal control and learning for visual navigation in novel environments. *arXiv preprint arXiv:1903.02531*, 2019. 2.2

[5] 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. 3.2

[6] Berta Bescos, José M Fácil, Javier Civera, and José Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018. 2.2

[7] Shehroze Bhatti, Alban Desmaison, Ondrej Miksik, Nantas Nardelli, N Siddharth, and Philip HS Torr. Playing doom with slam-augmented deep reinforcement learning. *arXiv preprint arXiv:1612.00380*, 2016. 2.2

[8] Sean Bowman, Nikolay Atanasov, Kostas Daniilidis, and George Pappas. Probabilistic data association for semantic slam. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2017. doi: http://www.dx.doi.org/10.1109/ICRA.2017.7989203. 2.2

[9] Tom Bruls, Will Maddern, Akshay A Morye, and Paul Newman. Mark yourself: Road marking segmentation via weakly-supervised annotations from multimodal data. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1863–1870. IEEE, 2018. 2.2

[10] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 2016. 2.2

[11] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019. 1.1, 3.1

[12] John Canny. *The complexity of robot motion planning*. MIT press, 1988. 2.2

[13] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *Conference on Robot Learning*, pages 947–956, 2018. 3.2

[14] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019. 3.1, 3.2

[15] Ming-Fang Chang, John W Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. (document), 1.1, 3.1, 3.1, 3.2, 3.3.4, 3.3.6, 3.4.1, **??**, **??**, 3.4.2, 3.5.2

[16] Changan Chen, Yuejiang Liu, Sven Kreiss, and Alexandre Alahi. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. *arXiv preprint arXiv:1809.08835*, 2018. 2.2

[17] Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning exploration policies for navigation. *arXiv preprint arXiv:1903.01959*, 2019. 2.2, 3, 2.4.2, 2.5.2

[18] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2.3.1

[19] Henggang Cui, Thi Nguyen, Fang-Chieh Chou, Tsung-Han Lin, Jeff Schneider, David Bradley, and Nemanja Djuric. Deep kinematic models for physically realistic prediction of vehicle trajectories. *arXiv preprint arXiv:1908.00219*, 2019. 3.1

[20] Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin, Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, and Nemanja Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2090–2096. IEEE, 2019. 3.2

[21] Nachiket Deo and Mohan M Trivedi. Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1179–1184. IEEE, 2018. 3.2

[22] Nachiket Deo, Akshay Rangesh, and Mohan M Trivedi. How would surround vehicles

move? a unified framework for maneuver classification and motion prediction. *IEEE Transactions on Intelligent Vehicles*, 3(2):129–140, 2018. 3.2

[23] Vinayak V Dixit, Sai Chand, and Divya J Nair. Autonomous vehicles: disengagements, accidents and reaction times. *PLoS one*, 11(12):e0168054, 2016. 2.5.2

[24] Nemanja Djuric, Vladan Radosavljevic, Henggang Cui, Thi Nguyen, Fang-Chieh Chou, Tsung-Han Lin, and Jeff Schneider. Motion prediction of traffic actors for autonomous driving using deep convolutional networks. *arXiv preprint arXiv:1808.05819*, 2018. 3.1, 3.2, 3.3

[25] Christian Dornhege and Alexander Kleiner. A frontier-void-based approach for autonomous exploration in 3d. *Advanced Robotics*, 27(6):459–468, 2013. 2

[26] Kuan Fang, Alexander Toshev, Li Fei-Fei, and Silvio Savarese. Scene memory transformer for embodied agents in long-horizon tasks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2.2

[27] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998. 2.2

[28] Dhiraj Gandhi, Lerrel Pinto, and Abhinav Gupta. Learning to fly by crashing. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3948–3955. IEEE, 2017. 2.2

[29] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. *arXiv preprint arXiv:2005.04259*, 2020. 1.1, 3.1, 3.2, 3.4.2, **??**

[30] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *CVPR*, 2017. 2.2

[31] Abner Guzman-Rivera, Dhruv Batra, and Pushmeet Kohli. Multiple choice learning: Learning to produce multiple structured outputs. In *Advances in Neural Information Processing Systems*, pages 1799–1807, 2012. 3.3.5

[32] Raia Hadsell, Pierre Sermanet, Jan Ben, Ayse Erkan, Marco Scoffier, Koray Kavukcuoglu, Urs Muller, and Yann LeCun. Learning long-range vision for autonomous off-road driving. *Journal of Field Robotics*, 26(2):120–144, 2009. 2.2

[33] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008. (document), 3.1

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2.4.2, 3.2

[35] Joey Hong, Benjamin Sapp, and James Philbin. Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8454–8462, 2019. 3.2

[36] Elia Kaufmann, Mathias Gehrig, Philipp Foehn, René Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scaramuzza. Beauty and the beast: Optimal methods meet learning for drone racing. *arXiv preprint arXiv:1810.06224*, 2018. 2.2

[37] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *RA*, 1996. 2.2

[38] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Lyft level 5 av dataset 2019. urlhttps://level5.lyft.com/dataset/, 2019. 3.1

[39] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *European Conference on Computer Vision*, pages 201–214. Springer, 2012. 3.2

[40] Henrik Kretzschmar, Markus Spies, Christoph Sprunk, and Wolfram Burgard. Socially compliant mobile robot navigation via inverse reinforcement learning. *The International Journal of Robotics Research*, 35(11):1289–1307, 2016. 2.2

[41] 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. 2.2

[42] Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998. 2.2

[43] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 336–345, 2017. 1.1, 3.2

[44] Cindy Leung, Shoudong Huang, and Gamini Dissanayake. Active slam in structured environments. In *2008 IEEE International conference on Robotics and Automation*, pages 1898–1903. IEEE, 2008. 2.2

[45] Junwei Liang, Lu Jiang, Kevin Murphy, Ting Yu, and Alexander Hauptmann. The garden of forking paths: Towards multi-future trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 4.2

[46] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 2.3.1

[47] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3.2

[48] Osama Makansi, Eddy Ilg, Ozgun Cicek, and Thomas Brox. Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future predic-

tion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7144–7153, 2019. 3.3.5, 3.4.2

[49] 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. 2.2

[50] Jean Mercat, Thomas Gilles, Nicole El Zoghby, Guillaume Sandou, Dominique Beauvois, and Guillermo Pita Gil. Multi-head attention for multi-modal joint vehicle motion forecasting, 2019. (document), 3.1, 3.2, 3.3, 3.3.2, 3.3.4, 3.4.2, **??**, 3.2, 3.5.2

[51] Atanas Mirchev, Baris Kayalibay, Patrick van der Smagt, and Justin Bayer. Approximate bayesian inference in spatial environments. *arXiv preprint arXiv:1805.07206*, 2018. 2.2

[52] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andy Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. Learning to navigate in complex environments. In *ICLR*, 2017. 2.2

[53] Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, and Vladen Koltun. Driving policy transfer via modularity and abstraction. *arXiv preprint arXiv:1804.09364*, 2018. 2.2

[54] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1):33–55, 2016. 2.2

[55] Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. Covernet: Multimodal behavior prediction using trajectory sets. *arXiv preprint arXiv:1911.10298*, 2019. 3.2, **??**, **??**

[56] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 3406–3413. IEEE, 2016. 2.2

[57] Nicholas Rhinehart, Kris M Kitani, and Paul Vernaza. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 772–788, 2018. 3.1

[58] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. Precog: Prediction conditioned on goals in visual multi-agent settings. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 3.2

[59] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer, 2016. 3.2

[60] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2.3.1, 2.4.2

[61] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3.2

[62] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. In *ICLR*, 2018. 2.4

[63] Manolis Savva, Angel X. Chang, Alexey Dosovitskiy, Thomas Funkhouser, and Vladlen Koltun. MINOS: Multimodal indoor simulator for navigation in complex environments. *arXiv:1712.03931*, 2017. 2.1

[64] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1.1

[65] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 3

[66] Rakesh Shrestha, Fei-Peng Tian, Wei Feng, Ping Tan, and Richard Vaughan. Learned map prediction for enhanced mobile robot exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*. IEEE, 2019. 2.2

[67] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 3.5.2

[68] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: An open dataset benchmark. *arXiv preprint arXiv:1912.04838*, 2019. 3.1

[69] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. 3.2

[70] Charlie Tang and Russ R Salakhutdinov. Multiple futures prediction. In *Advances in Neural Information Processing Systems*, pages 15398–15408, 2019. 3.1, 3.2, 3.3, 3.3, 3.3.1, 3.3.4

[71] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005. 2.2

[72] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3.5.2

[73] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 3.2, 3.3.3, 3.3.3

[74] Yi Wu, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian. Building generalizable agents with a realistic and rich 3d environment. *arXiv preprint arXiv:1801.02209*, 2018. 2.1

[75] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019. 3.2

[76] Marek Wydmuch, Michał Kempka, and Wojciech Jaśkowski. Vizdoom competitions: Playing doom from pixels. *IEEE Transactions on Games*, 2018. 2.1

[77] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018. 1.1, 2.1

[78] Brian Yamauchi. A frontier-based approach for autonomous exploration. 1997. 2.2, 2

[79] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8660–8669, 2019. 1.1

[80] Stephan Zheng, Yisong Yue, and Jennifer Hobbs. Generating long-term trajectories using deep hierarchical networks. In *Advances in Neural Information Processing Systems*, pages 1543–1551, 2016. 3.2

[81] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 633–641, 2017. 2.3.1

[82] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE, 2017. 2.2, 2.3.1