Learned Sampling Distributions for Efficient Planning in Hybrid Geometric and Object-Level Representations

Learned Sampling Distributions for Efficient Planning in Hybrid Geometric and Object-Level Representations

Katherine Liu*, Martina Stadler*, and Nicholas Roy

***Abstract*— We would like to enable a robotic agent to quickly and intelligently find promising trajectories through structured, unknown environments. Many approaches to navigation in unknown environments are limited to considering geometric information only, which leads to myopic behavior. In this work, we show that learning a sampling distribution that incorporates both geometric information and explicit, object-level semantics for sampling-based planners enables efficient planning at longer horizons in partially-known environments. We demonstrate that our learned planner is up to 2.7 times more likely to find a plan than the baseline, and results in up to a 16% reduction in traversal costs as calculated by linear regression. We also show promising qualitative results on real-world data.**

I. Introduction

We would like to enable intelligent autonomous navigation in structured, unknown environments. Algorithms that rely on dense geometric representations to generate motion plans often result in myopic behavior when deployed in novel, unknown environments. Consider a robot equipped with a RGB-D camera, with limited range and field of view. Under traditional planning paradigms, the robot often must first map an entire room, before realizing it must be exited via a door.

Object-level semantic information can provide important contextual cues beyond the range of dense geometric information to inform more intelligent long-horizon decisions. Not only do some studies suggest that object-level navigation is a biologically plausible motion-planning strategy [1], [2], many object-level elements in the environment can inform higher-level navigation decisions (i.e., traveling to a door is the most efficient way of exiting a room). With the advent of low-cost object detection [3], [4], various object-level mapping approaches have been proposed [5]–[7], enabling a renewed investigation of how to leverage object-level semantic cues for real-world robot navigation.

Despite their intuitive usefulness, object-level maps can be difficult to integrate within planners designed for unknown environments. Some approaches to navigation in unknown environments use generative models to predict unobserved occupancy information, such as [8], but are strongly coupled to well designed hierarchical representations for which relationships between objects may be hard to specify. Other approaches intelligently factor the environment into frontiers [9], but while doors are frontiers between rooms and hallways, not all objects should be navigation waypoints – exit signs are objects that provide information about nearby doors, and windows indicate the extents of buildings. Furthermore, the utility of object relationships can be a complex function that may change conditioned on the ultimate objective. For example, the usefulness of a door with an exit sign is determined by whether the goal is outdoors.

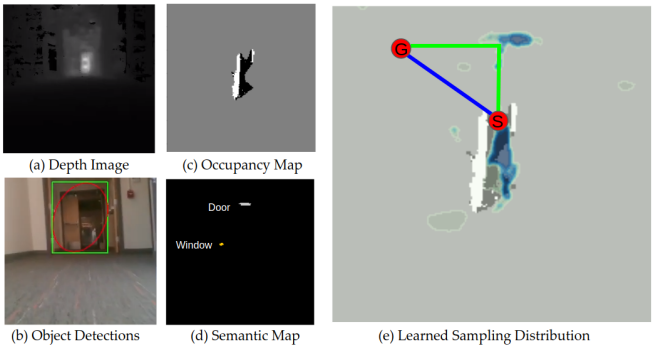

Fig. 1: In this work, we combine geometric (c) and object-level (d) representations built from depth images (a) and object detections (b) to predict sampling distributions for sampling-based motion planners (e). Instead of myopically assuming that unknown space is always traversable (blue path), our learned sampling distribution can be used to leverage contextual cues such as the existence of a door to exit the room (green path).

Modern sampling-based motion planners (SBMPs) such as the RRT* [10], PRM [11], etc., reduce computational intractability, but are generally optimized for dense geometry in fully known environments. While it has been shown that the efficiency of SBMPs can be dramatically improved by intelligent sampling strategies [12]–[14], the development of efficient sampling algorithms to speed up planning has largely focused on dense geometry. Examples include exploiting geometric properties of the current best solution [15], and learning sampling distributions using variational autoencoders [16] [17]. Furthermore, state of the art sampling algorithms tend to focus on improving planning speed or safety [18], but generally do not consider planning performance in unknown environments, where planners often only have access to incomplete geometric information. Although approaches such as [19] implicitly incorporate semantics into geometric representations, they do not reason over explicit object-level maps, which can potentially be useful for making intelligent decisions at a longer range.

In this work, we combine the computational power of randomized motion planners with higher-level semantic information via a learned sampling distribution, enabling intelligent navigation in structured, unknown environments.

*The first two authors contributed equally to this work.

All authors are with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology in Cambridge, USA. {katliu,mstadler,nickroy}@mit.edu

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-17-2-0181. Their support is gratefully acknowledged.

We propose optimizing a predictive sampling distribution inferred from dense geometric representations that track unobserved space, explicit object-level contextual cues both within and beyond the range of dense geometry, and information about the goal. This enables SBMPs to not only find plans quickly, but also reason about which plans are most likely to reach the goal despite incomplete geometric information. Rather than pre-specifying navigation rules and heuristics over our representations, we show that a Convolutional Neural Network (CNN) can be leveraged to synthesize multi-modal map information into a proper sampling distribution, and that learning a sampling distribution over geometric and semantic information improves navigation results in unknown environments.

II. Learned Sampling Distributions

In this section, we first review the definition of a planning problem, then re-formulate the problem in the context of learning a maximum-likelihood sampling distribution informed by multimodal information. Finally, we describe how to generate training data and the structure of the model.

A. Optimal Planning in Unknown Environments

The goal of motion planning is to find a feasible trajectory from an initial state to a terminal state. Let $\mathcal{X} \in \mathbb{R}^d$ denote the state space of the robot, where d is the cardinality of the state. It is also commonly assumed that $\mathcal{X}$ is composed of occupied and unoccupied states, $\mathcal{X}_{free}$ and $\mathcal{X}_{occ}$, and that there exists a known mapping from every $\mathcal{X}$ to its occupancy value, $f : \mathcal{X} \rightarrow \{\mathcal{X}_{free}, \mathcal{X}_{occ}\}$. In practice, when occupancy information is unavailable, the state is often assumed to be in $\mathcal{X}_{free}$. By denoting a trajectory to be a continuous mapping through $\mathcal{X}$, i.e., $s : [0,1] \rightarrow \mathbb{R}^d$, and a start and goal $x_s, x_g \in \mathcal{X}$, the planning problem can be formally stated as the minimization of some scalar cost function $\boldsymbol{c}(s) \in \mathbb{R}$ subject to constraints:

$$\begin{aligned} & s^* = \arg\min_s \boldsymbol{c}(s) \\ \text{s.t.} \quad & s^*(t) \in \mathcal{X}_{free} \; \forall t \in [0,1] \\ & s^*(0) = x_s, s^*(1) = x_g. \end{aligned} \tag{1}$$

Sampling-based motion planners solve this optimization problem by extracting random graphs [15] to approximate a solution to Equation 1. Graphs are generated by sampling states from a sampling distribution $P(x;\Gamma)$, where Γ are optional additional inputs to the distribution, then constructing a navigation graph G from valid states, where $G = \{V, E\}$, $V = \{v_0, v_1, ...v_n : v \sim P(x;\Gamma)\}$, $E = \{e_0, e_1, ...e_m\}$, and $V, E \in \mathcal{X}_{free}$. For notational convenience, we will also interchangeably express an edge as e_{ij} to represent the directed edge from v_i to v_j. In this representation, the n vertices of the graph correspond to the sampled states, and the m edges represent feasible traversals from one state to another[1]. A valid trajectory $\mathcal{T}$ is a directed, connected, acyclic sub-graph of G_t. The optimization in Equation 1 can then be formulated as:

$$\begin{aligned} & \mathcal{T}^* = \arg\min_{\mathcal{T}} \boldsymbol{C}(\mathcal{T}) \\ \text{s.t.} \quad & \mathcal{T}^* = \{V^*, E^*\}, s(e_0) = x_s, t(e_m) = x_g, \\ & v \neq v' \;\; \forall v, v' \in V^* \\ & t(e_i) = s(e_{i+1}) \; \forall i \in \{1...m-1\} \\ & V^* \in V, E^* \in E, \end{aligned} \tag{2}$$

where $\boldsymbol{C}(\mathcal{T}) \in \mathbb{R}$ is a scalar cost function that evaluates the cost of a sub-graph, $t(e)$ indicates the terminal node of edge e, $s(e)$ indicates the start node of edge e, and $V, E \in \mathcal{X}_{free}$.

Intuitively, a good choice of $P(x;\Gamma)$ is one more likely to sample Gs that contain $\mathcal{T}$s that are close to the solution to Equation 2. A common approach is to leverage a fully observed occupancy map and the geometric location of the goal (M_g^k and x_g) as conditioning parameters, $\Gamma = \{M_g^k, x_g\}$ [12], [19], [20]. However, this choice of parameterization is often limiting in real-world environments, where sensor limitations mean that geometric models are almost certainly inaccurate and incomplete. Assuming maps are complete is particularly detrimental when traditional methods rely on complete geometric information to eliminate low-cost but ultimately infeasible paths by searching only in the space of valid trajectories (i.e. $\{\mathcal{T} \mid \forall e \in \mathcal{T}, e \in \mathcal{X}_{free}\}$).

B. Multimodal Information for Navigation

In this work, we propose a novel approach to navigation in unknown environments, which learns a sampling strategy from both geometric and non-geometric navigational cues. Specifically, we would like to take advantage of sparse, metrically correlated object-level maps, which can provide semantic information both within and beyond the partial dense geometric map, to inform planning. By learning a predictive sampling distribution inferred from information about unknown space, low-level geometry, and explicit object-level contextual cues such as doorway and exit signs, we can empower SBMPs to reason about the pertinent structure of environments that have only been partially observed and therefore find better plans. For example, instead of having to densely map the walls of a room in order to exit it, the presence of a door is a clear semantic cue for a navigation strategy that has high probability of success (see Figure 1). Formally, we let $\Gamma = \{\mathcal{M}, \mathcal{C}\}$, where $\mathcal{M}$ denotes a *hybrid* form of map representation that incorporates low-level geometric information with semantic, object-level information (both of which may be incomplete). The semantic map is generated by projecting object-level maps onto a grid, where grid cell values indicate an object's class. $\mathcal{C}$ denotes planning context parameters, such as the goal location and whether the goal is indoors or outdoor. We solve Equation 2 over the graph sampled from $P(x;\{\mathcal{M}, \mathcal{C}\})$.

However, even with an optimized sampling distribution, we observe that simply being more likely to place samples along the optimal path does not encourage more likely trajectories to be chosen if the objective function of the graph search is not also informed by the information in the

[1]We observe that the vertex connection strategy varies between different algorithms, but neglect the notation here for readability.

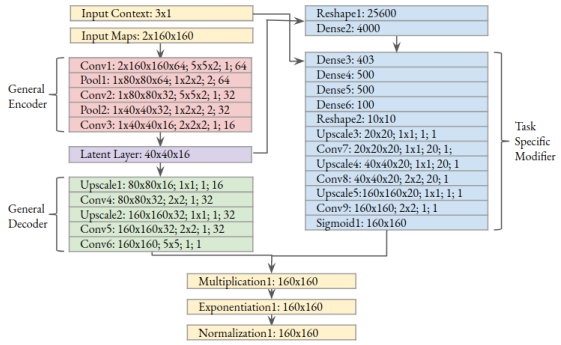

Fig. 2: Neural network used to generate learned sampling distributions. The network uses geometric and semantic information to learn general and task specific sampling distributions that empower a robot to reason about navigational modes in unknown space. Layer data indicates, in order, layer size, convolution size, stride size, and number of filters. Dropouts are not pictured but used for regularization. The context information is a vector with the first two values proportional the relative location of goal, and a scaled indicator variable indicating whether the goal is inside or outside.

hybrid map. It is easy to see that in the limit of infinite samples, an objective function that utilizes distance only would revert back to the myopic behavior of SBMPs using uniform sampling when operating in unknown environments. To mitigate this, we propose using the probability function not only to define the implicit planning graph, but *also* as a cost function over graph edges:

$$\boldsymbol{C}(\mathcal{T}) \propto \mathrm{P}(\mathcal{T}). \tag{3}$$

C. Learning Sampling Distributions

In general, determining the form of $P(x; \{\mathcal{M}, \mathcal{C}\})$ analytically is intractable. Instead, we would like to approximate the distribution with some mapping ϕ that depends on the hybrid map representation and the contextual information:

$$P(\mathcal{T}^*; \mathcal{M}_i, \mathcal{C}_i) \approx \phi(\mathcal{T}^*, \alpha; \mathcal{M}_i, \mathcal{C}_i). \tag{4}$$

where α denotes the parameters of the predictive sampling distribution model. A naive approach to the optimization of ϕ is to assume a dataset of optimal sampling distributions $P^*(\mathcal{T}^*)$, then attempt to minimize some distance metric between $P^*(\mathcal{T}^*)$ and $\phi(\mathcal{T}^*, \alpha; \mathcal{M}_i, \mathcal{C}_i)$. In practice, specifying $P^*(\mathcal{T}^*)$ is difficult, especially because such a distribution must account for the existence of multiple optimal trajectories that may exist in a complex environment.

While it is difficult to obtain example ground-truth sampling distributions, obtaining a representative dataset $\mathcal{D}$ of q example trajectories along with partial, hybrid environmental information and context information such that $\mathcal{D} = \{(\mathcal{T}_0, \mathcal{M}_0, \mathcal{C}_0), (\mathcal{T}_1, \mathcal{M}_1, \mathcal{C}_1), ..., (\mathcal{T}_q, \mathcal{M}_q, \mathcal{C}_q)\}$ is a far more tractable endeavor. Therefore, we propose learning a mapping for the sampling distribution by maximizing the probability that an optimal trajectory $\mathcal{T}^*$ will be sampled given $\mathcal{M}_i$ and $\mathcal{C}_i$, over the training dataset:

$$\arg\min_{\alpha} \sum_{i}^{q} -\log P(\mathcal{T}^*, \alpha; \mathcal{M}_i, \mathcal{C}_i), \tag{5}$$

where the maximum likelihood optimization has been converted into a negative log-likelihood minimization. The dataset $\mathcal{D}$ may be obtained offline in simulation, or in a gradual online fashion running any traditional planner that eventually finds optimal plans.

However, learning a joint distribution over V and E is extremely difficult due to the exponential state space of all possible sampled graphs. We therefore approximate Equation 5 as the joint distribution over nodes only:

$$\arg\min_{\alpha} \sum_{i}^{q} -\log P(V_i, \alpha; \mathcal{M}_i, \mathcal{C}_i). \tag{6}$$

Equation 6 abstracts trajectories into collections of nodes, rather than edges, and is similar to the approach taken in [16]. For computational tractability, we make a final approximation that the vertices in the sampled trajectory are conditionally independent:

$$\arg\min_{\alpha} \sum_{i}^{q} -\log(\prod_{j=0}^{|\mathcal{T}_i|} P(v_{ij}, \alpha; \mathcal{M}_i, \mathcal{C}_i)). \tag{7}$$

Finally, we substitute in our learned approximation ϕ for P:

$$\arg\min_{\alpha} \sum_{i}^{q} (\sum_{j=0}^{|\mathcal{T}_i|} -\log \phi(v_{ij}, \alpha; \mathcal{M}_i, \mathcal{C}_i)). \tag{8}$$

In practice, we optimize a scaled version of Equation 8 for numerical stability.

D. Neural Network Model Structure and Optimization

In this work, a convolutional neural network serves as the base form of model ϕ, and the optimization over α in Equation 8 is the optimization of the network weights. CNNs are particularly attractive due to their demonstrated successes in feature extraction, especially for multi-modal data inputs [21]. We exploit the modeling power of stacked convolutional layers to form our hybrid representation, i.e., $\mathcal{M} = [M_g; M_s]$ such that $\mathcal{M} \in \mathbb{R}^{qxqx2}$, where $M_g \in \mathbb{R}^{qxq}$ is a local pixel-wise map with occupancy information, $M_s \in \mathbb{R}^{qxq}$ is a discretized local semantic information map, and the two maps are concatenated together before being passed to the encoding CNN ϕ^{enc}, whose output is then decoded via ϕ^{dec} to form a general un-normalized distribution ϕ^{gen}:

$$\phi^{gen}(x_d; \alpha_{\{dec,enc\}}, \mathcal{M}) = \phi^{dec}_{x_d}(\alpha_{dec}, \phi^{enc}(\alpha_{enc}, \mathcal{M})), \tag{9}$$

where $q \in \mathbb{Z}$, $x_d \in \mathcal{X}_d$, $\mathcal{X}_d$ is a discretized form of $\mathcal{X}$, and $\alpha_{dec}, \alpha_{enc}$ denote the network weights for their respective networks. To incorporate contextual information with map-level information and obtain our final un-normalized distribution, we concatenate the output of the encoding CNN with $\mathcal{C}$, and pass the new matrix through a second decoding ϕ^{attn} that modifies the general distribution from Equation 9 to form our final unnormalized distribution ϕ^P, i.e.,

$$\begin{aligned}&\phi^P(x_d; \alpha_{\{dec,enc,attn\}}, \mathcal{M}, \mathcal{C}) = \\ &\phi^{gen}(x_d; \alpha_{\{dec,enc\}}, \mathcal{M}) \circ \phi^{attn}(x_d; \alpha_{\{attn,enc\}}, \mathcal{M}, \mathcal{C}),\end{aligned} \tag{10}$$

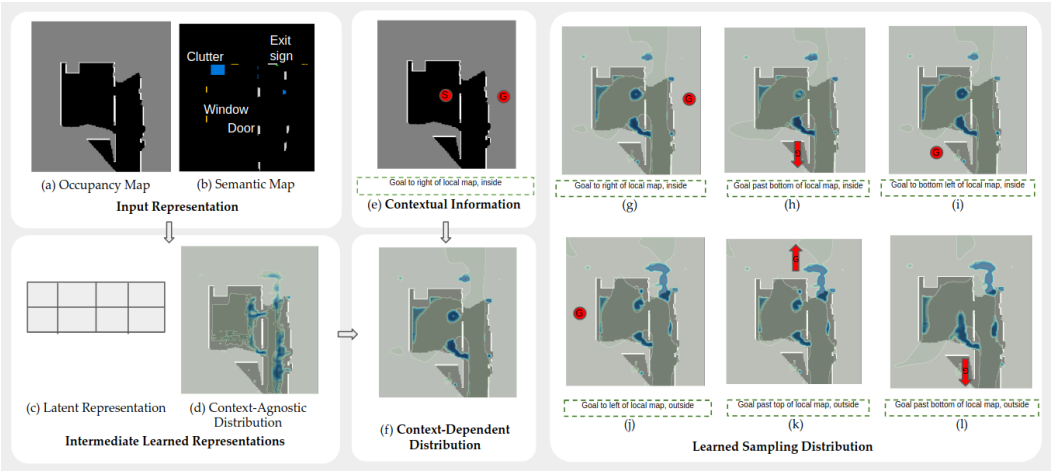

Fig. 3: Overview of model used to learn sampling distribution. Learned distributions are plotted as filled contour plots (blue is high probability, grey-green is low probability) overlaid on occupancy maps. In the first half of the network, local occupancy (a) and semantic (b) maps are passed through a CNN that learns a latent representation (c) and a context-agnostic distribution (d). A second network takes the latent layer (c) and contextual information (e) to learn a contextual attention matrix, which is multiplied against the general distribution to obtain a context-dependent distribution (f). The rightmost box shows the effect of different contextual inputs to the second network. Without contextual information, the distribution learns only general navigation heuristics. After adding contextual information about the goal, the network learns different navigation strategies. For example, when the goal is to the right and inside, the exit sign has low probability (g), but when the goal is to the left and outside, the exit sign has higher probability, biasing the planner to exit the building (j). The learned distribution is able to place probability near the exit sign, despite not having densely mapped the region.

where the context variable includes the relative location of the goal and a flag indicating whether the goal is indoors or outdoors, such that $\mathcal{C} \in \mathbb{R}^{d+1}$.

To enforce a proper probability distribution P, we pass the raw output of the network through an exponential, and then normalize the resulting positive output:

$$\phi(x_d; \alpha_{\{dec,enc\}}, \mathcal{M}, \mathcal{C}) = \frac{\exp(\phi_P(x_d; \alpha_{\{dec,enc\}}, \mathcal{M}, \mathcal{C})}{\sum_{x_d} \exp(\phi_P(x_d; \alpha_{\{dec,enc\}}, \mathcal{M}, \mathcal{C}))}. \quad (11)$$

We propose a two-stage optimization that learns two types of navigational modes separately. In the first stage, we optimize α_{enc} and α_{dec} and learn a task-independent sampling distribution that highlights general navigation modes of the environment. In the second stage, we freeze the network weights associated with general navigation modes, and learn a task-specific modifier of the original distribution that takes into account the context associated with a specific task (i.e., optimize α_{attn}). See Figure 2 for network details, and Figure 3 for qualitative outputs. The network optimizes Equation 11 via back-propagation and stochastic gradient descent with mini-batches.

E. Online Planning in Unknown Environments

To use the learned distribution online, we first optimize Equation 8 offline by applying Gradient Descent to the structure in Equation 11. During each timestep of online use, we generate incomplete local geometric and semantic maps, then run feed-forward prediction on our pre-trained model to recover a normalized probability distribution over the current, partially known map. Grid locations outside of the local sliding window are set to the minimum probability of the predicted window, and the sampling distribution over the entire map is again re-normalized. When sampling, we first sample from the discrete probability distribution, and then sub-sample within the discretization for a continuous sampling distribution over $\mathcal{X}$. Finally, we use a SBMP to generate a trajectory from the current robot position to the goal, but modify the planner to sample graph vertices from the learned distribution and to score graph edges using the objective defined in Equation 3.

III. Experiments

In this section, we describe the experimental procedure used to benchmark our learned sampling distribution against a uniform sampling distribution in simulated floorplan environments, and report aggregate metrics. Finally, we show qualitative results on data collected by a real-world vehicle.

A. Experimental Setup

We first demonstrated our approach in simulation using real-world floorplan data from MIT [22], which included floorplans from 13 buildings, split into 10 train buildings and 3 test buildings. The data was used to generate discretized occupancy and semantic maps, where door locations were provided by the dataset; plausible windows and exit signs were manually annotated. We assumed a holonomic vehicle with a planar depth sensor with an 85.2 degree field of view and 5 meter range and a RGB vision system with a 69.4 degree field of view capable of observing objects from 10 meters away, with no sensor or pose estimate noise.

B. Dataset Generation

We generated 1000 navigation tasks in known maps, categorized by indoor or outdoor start and goal locations, omitting outdoor only tasks due to the lack of semantic information outdoors in our environments. We simulated

Dataset	*II*	*II*	*II*	*II*	*IO*	*IO*	*IO*	*IO*
N	100	500	1000	5000	100	500	1000	5000
$\bar{C}$	0.84	0.95	0.92	0.90	0.95	1.11	1.01	0.99
$\|\|D\|\|$	1500	1162	1016	783	1500	550	705	595
$\|\|D_m\|\|$	642	819	787	669	140	132	229	15
R^2 Score	0.81	0.87	0.77	0.82	0.67	0.34	0.63	0.59
$\overline{C_l/C^*}$	1.21 $\pm$0.01	1.24 $\pm$0.01	1.31 $\pm$0.02	1.42 $\pm$0.03	1.22 $\pm$0.03	1.73 $\pm$0.16	1.75 $\pm$0.09	2.41 $\pm$0.09
$\overline{C_b/C^*}$	1.40 $\pm$0.02	1.32 $\pm$0.01	1.38 $\pm$0.02	1.51 $\pm$0.03	1.24 $\pm$0.02	1.51 $\pm$0.06	1.69 $\pm$0.07	2.35 $\pm$0.08

TABLE I: A comparison of plan costs between the learned sampling disribution and baseline planners when both planners succeed, broken out by dataset and planner iterations per query (N), which is correlated to the number of samples drawn per step. $||D||$ and $||D_m||$ are the total number of trials run and the number of trials where both planners succeeded. $||D||$ varies over N due to the time involved in running longer trials. $\bar{C}$ and R^2 score are the slope and fit score of a linear regression with zero intercept. $\overline{C_l/C^*}$ and $\overline{C_b/C^*}$ are the mean and standard error of the mean of the trajectory cost divided by the resolution-optimal trajectory cost for learned and baseline respectively. In the *II* dataset, the learned sampling distribution finds lower cost plans than the baseline, indicating that using the learned sampling distribution with a SBMP results in better navigation outcomes.

a robot completing each navigation task using a modified open source implementation of PRM [23] and a Euclidean objective function without *a priori* map information.

At each timestep, the robot generated and executed a plan, uncovering geometric and semantic information. We assumed that both maps were well approximated by raycasting in known maps according to the sensor characteristics. To generate diverse expert trajectories, at random intervals during the task we randomly selected a new goal location and solved for a dense, resolution optimal path to the goal location using a graph-based search over the ground-truth occupancy map with soft costs[2]. We then extracted body-centered local maps of 160x160 pixels (i.e., M_g, M_s) and label them with the points in the example trajectory. This dataset is then used to train the model in Equation 8. To examine the utility of our approach under different contexts, we create two evaluation datasets, the first where the robot begins indoors, and is told that the goal is also inside (*Inside-Inside*, or *II*), and another where the robot begins indoors, and is informed that the goal is outdoors (*Inside-Outside*, or *IO*). An evaluation dataset where the robot begins outdoors is not included, as our floorplans lack outdoor semantic information.

C. Simulation Evaluation Results

To evaluate our approach, we simulated a robot completing online navigation tasks in unknown environments using a modified version of PRM [23] with our learned sampling distribution and cost function. We compared PRM using our learned sampling distribution (*PRM + LSD*) to a baseline approach using a uniform sampling distribution (*PRM + Unf*) on a test set comprised of random start and goal locations over three previously unseen floorplans. Initial yaws were randomly generated per trial. *PRM + LSD* approximated Equation 3 for the search objective, letting the

[2]In practice, expert trajectories may be generated in many ways. We have assumed that these paths approximate optimal sub-graphs and can be represented as a set of edges and vertices.

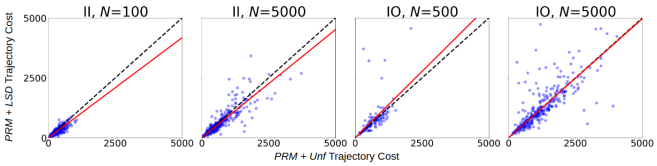

Fig. 4: Selected scatter plots of distance travelled in simulation units for *PRM + LSD* vs. *PRM + UNF* (blue circles) and the slope calculated by linear regression (red line) for various test conditions, where N is as in Table I. The dotted line is plotted as a reference for equal cost.

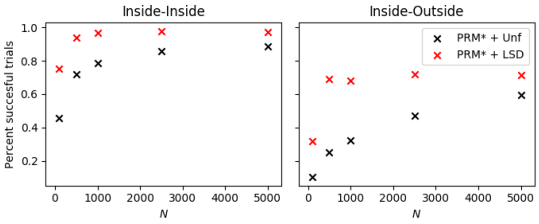

Fig. 5: A comparison of plan success rates between the learned (*PRM + LSD*, shown in red) and baseline (*PRM + Unf*, shown in black) planners, where N is as in Table I. The learned sampling distribution finds plans more frequently than the baseline, demonstrating that our learned sampling measure empowers PRM to find plans more quickly. We observe that the limitations placed on the timesteps per trial may inhibit the convergence of the harder *IO* test set. A small percentage ($< 2\%$) of trials were marked as failures due to the vehicle coming into collision with the environment; these trials were removed when calculating success statistics.

cost of an edge be an integral approximation of the negative log probability along an edge. *PRM + Unf* used a Euclidean search objective. We varied the maximum number of samples the respective planners were allowed to draw to generate a roadmap, and simulated trials for start and goal locations for which feasible trajectories existed. If no trajectory was found at a given timestep, or the simulation of a trial exceeded 10,000 timesteps, the entire trial was marked as a failure. Figure 3 shows qualitative results for the distribution trained without context, as well as the distribution after adding the context-based attention mechanism. The results demonstrate the utility of adding the semantic context of the goal. In the example shown, whether the exit sign is an area of high probability is related to goal being located inside or outside.

Plans generated by the learned sampling distribution and baseline are compared in Fig. 5, demonstrating that *PRM + LSD* was more likely to find feasible plans than *PRM + Unf*. We note that the learned distribution outperformed the baseline most notably at low sample counts, indicating that our method is especially useful for resource-constrained platforms. While the learned distribution was more successful in both test sets for all N, we observe the most significant improvements in the *IO* test set. For example, for the $N = 500$ case, *PRM + Unf* had a success rate of 25%, while *PRM + LSD* had a success rate of 69%.

We also demonstrate that our approach enables efficient navigation in certain domains. A comparison of plan costs is shown in Table I; Figure 4 includes scatter plots of trajectory costs for a selection of scenarios. For the *II* test set, we demonstrated between a 5% and 16% decrease in plan cost (determined by linear regression) when using the

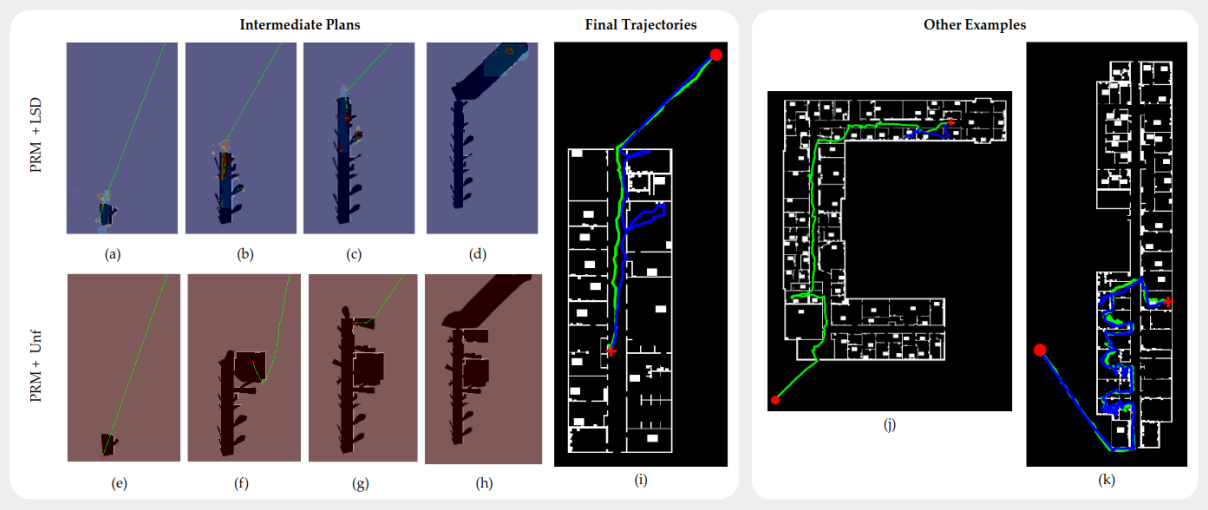

Fig. 6: Intermediate and final trajectories of the baseline and our method. In the left image, we show the sampling distribution (red is high probability, blue is low probability) overlaid on the robot's most recent occupancy map. The goal is set to the top right corner of the map, and the learned sampling distribution is also given the context that the goal is outdoors. Unlike the baseline (e)-(h), which uses only a Euclidean distance metric and greedily explores more rooms in the hopes of reaching the goal, the learned distribution largely encourages the planner to follow the hallway (a)-(d). Without the contextual information implicit in the learned sampling distribution, the baseline travels a much longer distance to reach the goal (i). In (i)-(k), we show the final trajectories of the learned (green) and baseline (blue) traversing from start (red cross) to goal (red circle). In (j), the baseline fails while trying to exit a room, but the learned largely remains in hallways and reaches the goal. In (k), both the learned and baseline mistakenly enter rooms prior to reaching the goal.

learned sampling distribution. In the *IO* test set, we generally demonstrated no statistically significant differences in plan cost for the two planners. In Figure 6-(a-i), we show a trial where the learned planner outperforms the baseline planner. The baseline planner greedily explores many rooms, while the learned planner places probability in the hallways and exit doors and navigates directly to the goal. In Figure 6-(j), we show a trial where the learned planner finds a near optimal plan and the baseline fails; we note that this trial, while clearly demonstrating the ability of the learned planner to find low cost trajectories, is not a mutually successful trial and therefore is not included in the aggregate statistics. In cases such as 6-(k), a suboptimal learned distribution causes *PRM + LSD* to suffer from failure modes similar to those of *PRM + Unf*, which leads to inefficient trajectories. Overall, we demonstrate that our learned sampling distribution significantly improves the success rate of PRM, and decreases plan costs over the *II* test set.

D. Real-World Evaluation Results

To evaluate the performance of a SBMP paired with the learned sampling distribution in a real-world environment, we performed qualitative analysis of planning perfomance on real-world data. The data was collected by a 1/10th scale racecar platform, carrying the Intel T265 and D435i modules (which provide state estimates and RGB-D images, respectively) and an Intel Nuc i7. Dense geometric maps were generating using a standard geometric estimation and mapping stack [24]. An object detector based on SSD-Mobilenet [3], [25], [26] was fine tuned to detect doors and windows with data from the OpenImages dataset [27], and an exit sign detector was written using HSV filtering. Object-level maps were generated by approximately projecting the output of an object mapping system [5] onto a plane. Figure 7 shows qualitative results from the experiment, demonstrating the utility of the learned sampling distribution.

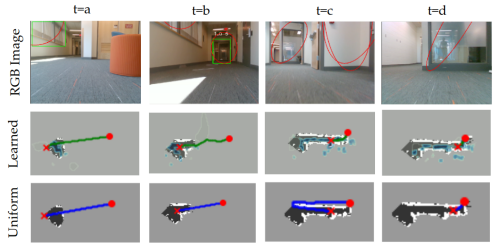

Fig. 7: Sensor data and plans at similar points on real-world data. RGB images show the object detections (green) and the estimated volume of objects projected into the image plane (red). The SBMP using the learned distribution generally plans to go down the middle of the hallway, while the SBMP using a uniform distribution greedily attempts to go through an unseen wall. In this case, the learned distribution empowers the SBMP to plan outside of the known geometric map by generally guiding the agent down the middle of the hallway, despite the noisy occupancy and object-level maps. All images are approximately aligned.

IV. Conclusion

We have presented a novel method for extending sampling-based motion planners into unknown environments by learning a sampling distribution. We demonstrate that sampling from the learned distribution in addition to utilizing the learned distribution as a cost function in a SBMP results in significantly higher success rates and can lower traversal costs in some domains. We have shown that learning a sampling distribution using object-level semantic information and geometric maps can enable long-horizon navigation in unknown environments, outperforming baseline, uniform sampling strategies. Finally, we have presented promising qualitative results on a real-world data.

References

[1] W. G. Chase, "Spatial representations of taxi drivers," in *The acquisition of symbolic skills*, Springer, 1983, pp. 391–405.

[2] M. F. Bonner and R. A. Epstein, "Coding of navigational affordances in the human visual system," *Proceedings of the National Academy of Sciences*, vol. 114, no. 18, pp. 4793–4798, 2017.

[3] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[4] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger.," in *2017 The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017, pp. 7263–7271.

[5] K. Ok, K. Liu, K. Frey, J. How, and N. Roy, "Robust object-based slam for high-speed autonomous navigation," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[6] V. Ila, L. Polok, M. Solony, and P. Svoboda, "SLAM++-a highly efficient and temporally scalable incremental slam framework," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 210–230, 2017.

[7] N. Sunderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid, "Meaningful maps with object-oriented semantic mapping," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[8] A. Pronobis, F. Riccio, and R. P. Rao, "Deep spatial affordance hierarchy: Spatial knowledge representation for planning in large-scale environments," in *ICAPS 2017 Workshop on Planning and Robotics*, 2017.

[9] G. J. Stein, C. Bradley, and N. Roy, "Learning over subgoals for efficient navigation of structured, unknown environments," in *Conference on Robot Learning*, 2018, pp. 213–222.

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

[11] L. E. Kavraki and J.-C. Latombe, "Probabilistic roadmaps for robot path planning," 1998.

[12] B. Burns and O. Brock, "Toward optimal configuration space sampling.," in *Robotics: Science and Systems*, Citeseer, 2005, pp. 105–112.

[13] M. Zucker, J. Kuffner, and J. A. Bagnell, "Adaptive workspace biasing for sampling based planners," 2008.

[14] R. A. Knepper and M. T. Mason, "Real-time informed path sampling for motion planning search," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1231–1250, 2012.

[15] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Batch informed trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 3067–3074.

[16] B. Ichter, J. Harrison, and M. Pavone, "Learning sampling distributions for robot motion planning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 7087–7094.

[17] B. Ichter and M. Pavone, "Robot motion planning in learned latent spaces," *IEEE Robotics and Automation Letters*, 2019.

[18] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *Ieee access*, vol. 2, pp. 56–77, 2014.

[19] I. Baldwin and P. Newman, "Teaching a randomized planner to plan with semantic fields," *TAROS 2010*, p. 20, 2010.

[20] A. Yershova, L. Jaillet, T. Simeon, and S. M. LaValle, "Dynamic-domain RRTs: Efficient exploration by controlling the sampling domain," in *2005 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2005, pp. 3856–3861.

[21] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[22] E. Whiting, J. Battat, and S. Teller, "Topology of urban environments," in *Computer-Aided Architectural Design Futures (CAADFutures) 2007*, Springer, 2007, pp. 114–128.

[23] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012, http://ompl.kavrakilab.org. DOI: 10.1109/MRA.2012.2205651.

[24] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *European conference on computer vision*, Springer, 2016, pp. 21–37.

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

[27] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, *et al.*, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *arXiv preprint arXiv:1811.00982*, 2018.