CogniPlan: Uncertainty-Guided Path Planning with Conditional Generative Layout Prediction

Yizhuo Wang[1] **Haodong He**[2] **Jingsong Liang**[1] **Yuhong Cao**[1†]
Ritabrata Chakraborty[3] **Guillaume Sartoretti**[1]
[1]National University of Singapore [2]Tongji University [3]BITS Pilani

Abstract: Path planning in unknown environments is a crucial yet inherently challenging capability for mobile robots, which primarily encompasses two coupled tasks: autonomous exploration and point-goal navigation. In both cases, the robot must perceive the environment, update its belief, and accurately estimate potential information gain on-the-fly to guide planning. In this work, we propose CogniPlan, a novel path planning framework that leverages multiple plausible layouts predicted by a conditional generative inpainting model, mirroring how humans rely on cognitive maps during navigation. These predictions, based on the partially observed map and a set of layout conditioning vectors, enable our planner to reason effectively under uncertainty. We demonstrate strong synergy between generative image-based layout prediction and graph-attention-based path planning, allowing CogniPlan to combine the scalability of graph representations with the fidelity and predictiveness of occupancy maps, yielding notable performance gains in both exploration and navigation. We extensively evaluate CogniPlan on two datasets (hundreds of maps and realistic floor plans), consistently outperforming state-of-the-art planners. We further deploy it in a high-fidelity simulator and on hardware, showcasing its high-quality path planning and real-world applicability.

Keywords: Path Planning under Uncertainty, Map Prediction, Graph Attention

1 Introduction

Path planning is a fundamental capability of robotic intelligence, enabling mobile robots to compute efficient routes toward designated targets to accomplish their missions. In practical scenarios, however, environments are usually unknown beforehand, posing inherent challenges for decision-making. This work focuses on two essential and coupled tasks: autonomous exploration and point-goal navigation, where the robot must plan in unknown environments to either fully map the space or reach a specific target, both via the shortest possible routes.

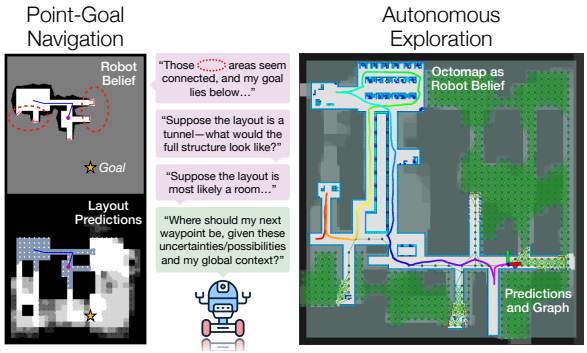

Figure 1: **CogniPlan's layout prediction and trajectory.** We show halfway navigation in a simulated map and halfway exploration in a Gazebo environment.

To reason effectively under such uncertainty in unknown environments, the robot must act on its partial and dynamically updating belief. At the same time, it must estimate potential information gain, where seemingly ambiguous choices may lead to significantly different long-term outcomes. However, most planning policies remain rule-based, typically selecting frontiers (i.e., the boundary between known traversable and unknown regions) using greedy heuristics based on proximity and/or

[†]Corresponding author. Project page: https://yizhuo-wang.com/cogniplan

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

expected unmapped volume [1–4]. While effective in simple scenarios, such approaches react passively to observations without actively reasoning about or predicting the broader environmental layout, which limits their ability to plan with non-myopic spatial coherence. Data-driven approaches, on the other hand, include deep reinforcement learning (DRL) methods that implicitly encode spatial knowledge within policy networks [5–9], as well as those explicitly predicting layouts to then apply conventional planning [10–14]. However, these two paradigms, implicit policy learning and explicit layout prediction, are rarely integrated, leaving their potential synergy underexplored. This work addresses the question: Can layout predictions be *explicitly* harnessed to guide planning under uncertainty, so that a learned planner can reason over multiple plausible spatial hypotheses?

When humans plan paths, their brain typically constructs a **cognitive map**: an internal spatial representation of the environment that supports reasoning, memory recall, and forward planning during navigation [15]. Inspired by this capability, we introduce **CogniPlan**: a fully learned path planning framework guided by multiple plausible layout predictions generated through conditional generative inpainting. We illustrate this cognitive reasoning process in Fig. 1, where our robot first explicitly predicts the layout structure and then implicitly reasons about and plans its path under uncertainty.

We first adopt a Wasserstein generative adversarial network (WGAN) [16] to generate plausible predictions conditioned on ground-truth layout types. We tailor our WGAN to be lightweight, allowing efficient repeated inference for generating diverse predictions that capture layout uncertainty. The output is further post-processed to produce binary outputs indicating either free space or obstacles. During forward inference, we input a set of layout conditioning vectors to generate multiple plausible predictions, each corresponding to a different structural assumption about the partially observed environment. Guided by these layout hypotheses, CogniPlan's graph attention network (inspired by [7]) reasons about potential uncertainty and information gain, allowing the agent to make real-time, informed decisions about the next waypoint for exploration or navigation.

Recent studies highlight graph attention-based approaches as flexible and scalable solutions for path planning, capable of generalizing across environments of varying sizes while maintaining a global context [7, 17]. However, since these policies usually rely solely on graph representations, they often fails to capture subtle yet meaningful details in the underlying occupancy map. For example, frontiers with distinct shapes are encoded similarly in the graph, despite conveying critical structural information. CogniPlan addresses this limitation by leveraging a generative model to predict plausible layouts, effectively capturing non-trivial spatial details and amplifying them at the graph level to guide informed planning decisions. We demonstrate a strong synergy between the scalability and reasoning capacity of the graph attention network and the high-fidelity spatial features captured by the generative model. Compared to using our layout generator with conventional planners [18–20] and to state-of-the-art DRL planners [7, 9], CogniPlan achieves 17.7% and 7.0% reduction in travel length for exploration, and a 3.9% and 12.5% reduction for navigation, respectively, across hundreds of simulated maps. We further evaluate CogniPlan on a realistic floor plan dataset, exhibiting higher efficiency without any additional training or fine-tuning. Finally, we deploy our framework in a high-fidelity Gazebo simulation (in Appendix A), as well as on a mobile ground robot, demonstrating its high-quality path planning and real-world applicability.

2 Related Works

Exploration Planner. Classical exploration relies on hand-crafted heuristics. A seminal example is frontier-based exploration [1], where robots navigate to proximal frontiers to gradually uncover unknown space. Later work improved efficiency by integrating information gain metrics like frontier size and expected unmapped volume [21–23]. However, basic frontier-based algorithms are inherently greedy, often causing inefficient backtracking and myopic behavior in complex environments [24]. Sampling-based methods, rooted in the next-best-view paradigm [25], alleviate these issues by optimizing longer-term trajectories [3, 26]. Advanced planners such as TARE [18, 19], HPHS [27], and FUEL [28] leverage hierarchical structures to balance global planning and local refinement. Learning-based methods typically use convolutional networks to encode the robot's envi-

ronmental belief [5, 6, 29, 30], preserving local geometry but suffering from limited receptive fields, which often forces planners to downsample or crop large environments. Recent work therefore looked to graph attention networks as backbones to further boost reasoning and scalability [7, 31]. However, graph-based representations remain constrained to node-level features, overlooking fine-grained, high-fidelity spatial details critical for effective planning.

Navigation Planner. Navigation planners can be broadly categorized into three groups: search-based, sampling-based, and learning-based methods. Search-based methods (e.g., A* [20], D* [32], and D*Lite [33]) rely on graph expansion with heuristic functions, while sampling-based methods (e.g., RRT [34], RRT* [4], and BIT* [35]) explore the continuous space through incremental random sampling. Despite their success in fully known environments, both categories struggle in unknown/partially known environments due to limited adaptability and lack of foresight. On the other hand, learning-based methods, especially DRL-based, offer the potential to reason over uncertainty and plan efficiently without hand-crafted rules [9, 36–38]. However, most navigation planners, similar to exploration ones, lack explicit spatial prediction of unknown regions, which limits their goal-directed behavior in complex scenarios [39]. In this work, we explicitly model and leverage such predictions to improve planning efficiency and long-horizon navigation under uncertainty.

Planning with Map Prediction. Predicting unknown layouts from observed environmental beliefs has emerged as a promising approach for exploration [10, 11, 14, 30, 40, 41], navigation [39, 42], or both [12, 43]. The use of deep learning methods, particularly generative models [16, 44], enables the prediction of semantically coherent and structurally plausible layouts, going beyond merely optimizing for statistical similarity to ground truth. Most existing methods rely on a *single* deterministic layout prediction without modeling uncertainty and decouple prediction from planning. They first generate a layout prediction and then compute information gain (e.g., predicted sensor coverage) to directly guide either a conventional planner [10, 11] or a learning-based one [30, 43]. More recent exploration planners such as UPEN [12], MapEx [14], and PIPE [41], as well as the navigation planner in [42], incorporate uncertainty estimates from learned predictors to support more informed decision-making by *conventional* planners. CogniPlan bridges prediction and planning by providing a fully learning-based, uncertainty-aware framework for both exploration and navigation. It estimates global layout uncertainty using multiple conditioning priors, which capture nuanced layout cues and amplify those important to downstream planning. While methods like MapEx rely on a few predictions without ensuring full exploration, CogniPlan guides planning with its predictions while maintaining coverage, making it more broadly applicable in real-world scenarios.

3 Framework Design

An overview of the CogniPlan framework is presented in Fig. 2, illustrating how the conditional generative inpainting network collaborates with the graph attention planner network. In this section, we explain how the fine-grained belief map $\mathcal{M}$, which contains non-trivial structural details/cues, is amplified through the layout predictions $\hat{\mathcal{M}}$. Information relevant to planning, including frontiers, connectivity, and uncertainty guidance, is extracted and encoded into a graph representation $G' = (V', E)$ for use in downstream planning.

3.1 Problem Formulation

We consider a bounded environment $\mathcal{E}$, partitioned into free space $\mathcal{E}_f$ and occupied space $\mathcal{E}_o$. The robot holds a spatial belief over the environment, which is discretized into a 2D occupancy map $\mathcal{M} \subset \mathbb{Z}^2$ for layout prediction and planning. Following a policy π, the robot navigates within the traversable free space, forming a collision-free trajectory $\psi : \{0, 1, 2, ...\} \rightarrow \mathcal{E}_f$. Here, $\psi_i(t)$ indicates the robot's waypoint location at timestep t. Upon observation with its onboard sensor (a LiDAR in our case) along ψ, the unknown areas $\mathcal{M}_u$ are gradually revealed and classified into either free areas $\mathcal{M}_f$ or occupied area $\mathcal{M}_o$, such that $\mathcal{M} = \mathcal{M}_u \cup \mathcal{M}_f \cup \mathcal{M}_o$. The exploration task is considered complete upon the closure of $\mathcal{M}_o$ with respect to $\mathcal{M}_f$, while the navigation task

is complete when the robot reaches the destination $\bar{p} \in \mathcal{E}_f$. Both planning tasks are formulated to minimize the cost of completion, measured by either travel length $D(\psi)$ or makespan.

3.2 Conditional Generative Inpainting

Image inpainting is a well-established technique for predicting missing regions in an image based on the surrounding context. Most existing methods [10–12, 43] utilize encoder-decoder architectures, such as variational autoencoder (VAE) [45] or U-Net [46], to generate local patches within occluded areas, limiting their use to refine the map and estimate information gain. We hypothesize that, with a diverse set of layout predictions, even if individually inaccurate, a planner network can gradually learn to balance exploration and exploitation. This allows it to reason about both potential information gain and the environment's topology, enabling non-myopic path planning.

Learning to Inpaint Layouts. To generate diverse predictions, we train a WGAN with gradient penalty on a dataset comprising three subsets of maps (namely room, tunnel, and outdoor), each paired with a ground-truth layout type conditioning vector encoded as one-hot during training. This layout conditioning vector z is then expanded to match the spatial dimensions of the input and concatenated with the robot's occupancy map belief and a mask indicating the unknown regions to be predicted. The generator outputs a layout prediction $\hat{\mathcal{M}} = \mathrm{Gen}(\mathcal{M} \mid z)$ that is composed of $\hat{\mathcal{M}}_f$ and $\hat{\mathcal{M}}_o$, and is trained with the following objective weighted by λ:

$$\mathcal{L}_{\mathrm{Gen}} = -\mathbb{E}\big[\mathrm{Dis}(\hat{\mathcal{M}})\big] + \lambda_1 \big\|\hat{\mathcal{M}} - \mathcal{E}\big\|_1 + \lambda_2 \big\|(\hat{\mathcal{M}} - \mathcal{E}) \odot M_{\mathrm{sd}}\big\|_1 / \sum M_{\mathrm{sd}} - \lambda_3 \,\mathrm{F1}(\hat{\mathcal{M}}, \mathcal{E}), \quad (1)$$

which includes one adversarial loss from the discriminator and three reconstruction terms, including an L1 loss over the entire prediction and a Dice (F1-score) loss for overlap accuracy. However, overemphasizing reconstruction loss in all unknown areas can mislead the training process. Intuitively, missing pixels near known regions are less ambiguous and more critical for ensuring robot traversability. Following [47], we implement a spatially discounted mask M_{sd} and extend it to a more flexible formulation that accommodates irregular mask shapes (known areas in $\mathcal{M}$). We assign higher weights to regions closer to known areas and apply a normalization term to account for varying mask sizes. The weights decay exponentially with distance from known regions, controlled by a decay factor (we use $\gamma = 0.95$). See Appendix B for more implementation details.

Layout Uncertainty during Inference. To account for the unknown layout type during inference, we design a set of diverse conditioning vectors $\mathcal{Z} = \{z_0, z_1, ...\}$, where each vector represents a specific structural pattern or a mixture thereof. This set includes both one-hot and soft one-hot variants, allowing the generator to produce a diverse mixture of layout predictions. We post-process the output to obtain binary predictions and apply morphological closing and opening operations to fill small holes and remove minor noises/artifacts. Since each pixel is classified as either free or occupied, the average of multiple predictions can fully characterize the uncertainty, as it follows a Bernoulli distribution. The inpainting model captures and amplifies non-trivial spatial cues in the occupancy map that are often overlooked by the graph representation, such as incomplete corridor connections, newly revealed wall corners detected by LiDAR, and multi-branch frontier structures emerging ahead of corridors. We then incorporate the estimated uncertainty into the graph representation used by our planner. Notably, our lightweight generator has fewer than 0.35M parameters, enabling real-time multiple forward inferences for uncertainty estimation, even when running entirely on CPU. Its efficiency also facilitates the use of the generated layout predictions for subsequent downstream planner model training without introducing much computational overhead, in contrast to the finetuned LaMa inpainting model [44] used in [14, 41], which has over 50M parameters.

3.3 Uncertainty-Guided Planner Network

Graph as Observation. Learning to plan on graphs offers a more scalable backbone than directly planning on occupancy map images, enabling better reasoning and reduced spatial complexity [7, 48]. Instead of building a graph from the occupancy map belief $\mathcal{M}_t$, we construct a collision-free graph $G_t(V_t, E_t)$ at each decision timestep t, based on the layout predictions $\hat{\mathcal{M}}_t$. We choose to uniformly distribute the nodes $v \in V$ over the free areas, and connect them based on proximity and

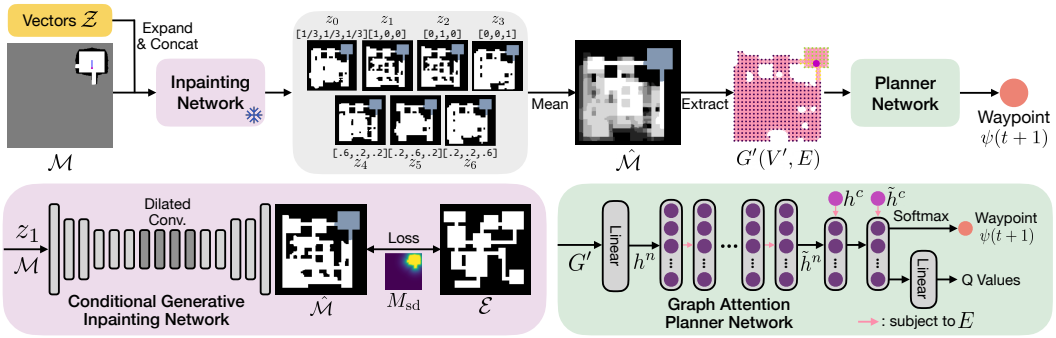

Figure 2: **CogniPlan framework.** We first train a generative inpainting network on procedurally-generated maps, given their ground-truth layout type vector (room, tunnel, or outdoor), and then freeze the model to train a graph-attention-based planner network. Our planner reasons over multiple predictions generated from a set of conditioning vectors by incorporating probabilistic information into the graph feature, and iteratively outputs the next waypoint for exploration or navigation.

line-of-sight visibility, forming edges $(v_i, v_j) \in E$. In addition to the node coordinates, we associate each node with uncertainty and frontier-related attributes into the graph, yielding augmented node representations $v_i' = (v_i, s_i, p_i, u_i, g_i) \in V'$. Specifically, **(1)** $s_i \in \{0, 1\}$ is the signal indicating whether the node lies in known or predicted areas; **(2)** $p_i \in [0, 1]$ is the average predicted probability of free areas. Note that the layout predictions are averaged to represent uncertainty; **(3)** $u_i \in [-1, 1]$ is the normalized utility, based on the number of observable frontiers at the node position (with $u_i = -1$ assigned in predicted areas); and **(4)** $g_i \in \{0, 1\}$ is the guidepost, indicating whether the node lies on the shortest trajectory to the nearest frontier within the known areas, which we find particularly useful for backtracking when the robot reaches a dead-end. Note that we compute the guidepost on the predicted graph rather than solely within the known areas, which helps optimize the path toward the frontier. Following [9], for the navigation task, we additionally incorporate a direction vector toward the point-goal into the graph. In this way, features from the predicted layout are encoded into an informative graph $G_t'(V_t', E_t)$, which is then fed into the planner network.

Policy Network. Our graph attention-based policy network comprises an encoder and a decoder: the encoder aggregates information from nodes across the known map, while the decoder selects the next neighboring node to visit based on the encoded global context.

The encoder consists of a feed-forward layer followed by N stacked masked self-attention layers. We first project the node properties in V_t' into d-dimensional features h^n, which are then enhanced through attention layers that aggregate information from neighboring nodes, subject to the mask derived from graph connectivity E_t. The mask constrains each node's attentive field to immediate neighbors, while stacking multiple layers progressively expands the context to N-hop neighborhoods. This enables the encoder to integrate both local structural information and broader spatial dependencies into a rich, uncertainty-aware representation $\tilde{h}^n$ of the graph/environment information. The decoder determines the final policy based on $\tilde{h}^n$. We first select the feature at the robot's current position as the query, $h^c = \tilde{h}^{\psi(t)}$, and use it with neighboring node features (as key-value pairs) in an attention layer to produce an updated feature $\tilde{h}^c$. A single-head attention mechanism is then applied, using $\tilde{h}^c$ and its neighboring node features, with the resulting attention scores directly serving as the output policy π [49], from which the next waypoint is selected. It allows the policy to adapt to varying numbers of neighbors, accommodating arbitrary graph connectivity.

Critic Network. The critic network estimates the state-action/Q values based on the robot's partial observations, providing learning signals to guide the policy network. We adopt privileged learning by directly feeding the ground-truth graph $G^*(V^*, E^*)$, extracted from $\mathcal{E}$, into the critic to improve value estimation, thereby facilitating more effective policy learning under a stochastic partially observable Markov decision process (POMDP). The critic has an identical network architecture as the policy network, except that its final layer outputs Q-values instead of single-head attention scores.

Similarly, the critic contains prediction probability information p_i as input, but only for nodes v_i^* in the ground-truth graph.

3.4 Network Training Details

Inpainting Model. Generative adversarial networks are prone to training instability, often resulting in issues such as mode collapse or gradient explosion. To mitigate this, we adopt a WGAN with gradient penalty (WGAN-GP [16]) and apply a warmup phase during the initial 10k iterations, where only the generator is trained using a pure reconstruction loss (last three terms in Eq. 1). This warmup encourages the generator to produce coarse but meaningful layout estimates before engaging in full adversarial training for the subsequent 490k iterations. The discriminator loss is given by:

$$\mathcal{L}_{\text{Dis}} = \mathbb{E}_{\hat{\mathcal{M}}}\big[\text{Dis}(\hat{\mathcal{M}})\big] - \mathbb{E}_{\mathcal{E}}\big[\text{Dis}(\mathcal{E})\big] + \lambda_{\text{gp}}\,\mathbb{E}_{\tilde{\mathcal{M}}}\left[\big(\|\nabla_{\tilde{\mathcal{M}}}\text{Dis}(\tilde{\mathcal{M}})\|_2 - 1\big)^2\right], \tag{2}$$

where $\tilde{\mathcal{M}} = \alpha\hat{\mathcal{M}} + (1-\alpha)\mathcal{E}$ is a random linear interpolation between layout prediction and the ground truth, with $\alpha \sim \mathcal{U}(0,1)$. The gradient penalty term, weighted by λ_{gp}, encourages the 1-Lipschitz constraint on the discriminator, following [16]. To build our training dataset for the inpainting network, we use TARE [18, 19] to explore the Dungeon environment provided by Chen et al. [50]. We procedurally generate 3000 random maps, evenly distributed across three layout types: room, tunnel, and outdoor environments. During exploration, we monitor the exploration rate and save the occupancy map whenever the rate increases by more than 10%.

Planner Model. Both the exploration and navigation planners are trained using the same inpainting model. We train the exploration planner on another set of 6000 maps (2000 per environment type) and the navigation planner on 2000 maps with assigned start and target positions, using the soft actor critic (SAC) algorithm [51] with discrete actions. Each map has a grid size of 250×250, with a sensing range of 40 pixels. Nodes are uniformly distributed at intervals of 9px and connected to neighboring nodes within a radius of 26px (connect to 25 neighbors at most in this setting). Empirically, we use a set of seven layout conditioning vectors (i.e., $|\mathcal{Z}| = 7$), consisting of one uniform vector, three one-hot vectors, and three soft-biased vectors favoring each layout type. Alternatively, our planner network can also be trained using a reduced set of four vectors or even a single uniform vector, although this leads to degraded performance. Additional details are provided in Appendix B.

4 Experiments

In this section, we evaluate CogniPlan's exploration performance across two datasets and deploy it on real robot using the same trained model without any retraining or fine-tuning. We also evaluate its navigation capability on simulated maps and compare both results against other baseline algorithms.

4.1 Evaluation on Simulated Maps

We evaluate CogniPlan on 150 maps for the exploration task, covering three different layout types in simulated environments, and on 100 maps for the navigation task. All evaluation maps are unseen during training. We measure and compare the travel length required to fully explore the environment or reach a designated point-goal. We compare CogniPlan against baseline exploration planners: **(1)** nearest: navigate to the nearest frontier; **(2)** utility: navigate to a frontier balancing proximity and utility; **(3)** NBVP [3]: a sampling-based exploration planner; **(4)** TARE Local [18, 19]: the local sampling-based planner from TARE; **(5)** ARiADNE+ [7]: an ablative improved ARiADNE variant with privileged learning and the same guidepost as CogniPlan. **(6)** Inpaint+TARE: an ablative baseline applying the TARE coverage planner to inpainted predictions treated as ground truth. For navigation, we compare against: **(1)** BIT [35]: a sampling-based navigation planner; **(2)** D*Lite [33]: a search-based navigation planner; **(3)** CA [9]: an ablative context-aware DRL navigation planner; **(4)** Inpaint+A*: an ablative method that uses our layout prediction and plans A* path to the closest unknown (if any) region to the goal. For all results, we use CogniPlan with four layout predictions (i.e., $|\mathcal{Z}| = 4$), from exploration and navigation planner model trained with seven, to balance performance and computational efficiency.

Table 1: **Exploration performance comparison over 150 maps (50 per environment).** We report the mean and standard deviation of travel length to complete exploration (lower is better).

	Nearest	Utility	NBVP	TARE Local	ARiADNE+	Inpaint+TARE	CogniPlan
Room	412.2(±51.3)	397.1(±63.3)	409.5(±63.4)	400.2(±55.3)	379.5(±54.5)	436.5(±109.0)	**360.1**(±43.5)
Tunnel	297.8(±69.3)	293.7(±73.6)	303.5(±87.1)	300.4(±75.2)	278.4(±66.7)	319.7(±85.3)	**268.5**(±58.9)
Outdoor	372.7(±67.6)	358.6(±60.3)	318.8(±72.9)	310.9(±56.0)	330.1(±52.9)	360.9(±73.0)	**290.7**(±50.8)
Average	360.9	349.8	343.9	337.2	329.3	372.3	**306.4**
Gap	15.1%	12.4%	10.9%	9.1%	7.0%	17.7%	0%

Table 2: **Navigation performance comparison over 100 maps.** We report the mean and standard deviation of travel length to reach the point goal (lower is better).

	BIT	D*Lite	CA	Inpaint+A*	CogniPlan
Distance	275.2(±130.4)	224.6(±85.9)	253.9(±112.4)	231.1(±125.8)	**222.1**(±64.7)
Gap	19.3%	1.1%	12.5%	3.9%	0%

We present the exploration results in Table 1 and the navigation results in Table 2, where CogniPlan consistently outperforms all baselines, including state-of-the-art learning-based approaches, and slightly surpasses the near-optimal navigation planner D*Lite. Compared to using inpainting with a conventional planner or learning-based methods without layout predictions (e.g., ARiADNE+ and CA), CogniPlan achieves a travel length reduction of 17.7% and 7.0% for exploration, and 3.9% and 12.5% for navigation. Interestingly, Inpaint+TARE performs particularly poorly. Upon closer inspection, we believe this may stem from high variability in layout predictions during the early stages of exploration, which causes the coverage planner to generate inconsistent and divergent trajectories. The planner struggles to effectively handle uncertainty and becomes overly reliant on potentially inaccurate predictions, often leading to zig-zagging paths and reduced efficiency. We believe this highlights the importance of using an uncertainty-aware downstream planner that can leverage layout uncertainty to guide more consistent and efficient planning.

To investigate the impact of the number of predictions, we evaluate CogniPlan using $|\mathcal{Z}| = 1, 4, 7$ predictions for both exploration and navigation tasks. As shown in Fig. 3, we report the reduction in travel length when using four or seven predictions compared to a single prediction. The layout types apply only to exploration. We observe that increasing the number of predictions during inference generally improves performance. However, the gains tend to saturate as the number of predictions increases, suggesting that a small number of diverse hypotheses is sufficient to guide planning.

4.2 Evaluation on Realistic Floor Plans

To benchmark CogniPlan in a more realistic setting, we further evaluate it on the KTH floor plan dataset [52] and on Gazebo environments (Appendix A), which, unlike standardized simulated maps suited for large-scale training and testing, better reflects real-world indoor environments. We compare CogniPlan against the following representative baseline methods in a zero-shot setting: **(1)** nearest; **(2)** ARiADNE+; **(3)** MapEx [14], a non-learning approach that uses map predictions and is trained on the KTH dataset to estimate information gain, representing the state-of-the-art for fast exploration on these maps. For fairness, we tune the parameters of all methods to maximize their performance on those maps and ensure consistency with those used in CogniPlan (where applicable).

We first deploy each method across 10 diverse floor plans, running it twice on each map with different starting positions. We record the exploration rate and travel length $D(\psi)$, and report the average exploration progress curves in Fig. 4. We observe that MapEx achieves the best performance during the early stages of exploration (up to 80%), but its progress plateaus when approaching full coverage. This is likely because MapEx prioritizes waypoints that are likely to yield high information gain; however, its effectiveness diminishes in the final stages, where the remaining unexplored regions tend to be smaller, more isolated, or harder to predict. Although these out-of-distribution layouts differ significantly from the data used to train CogniPlan's inpainting and planner models, leading to some predictions that visibly deviate from the ground truth, CogniPlan still maintains a steady ex-

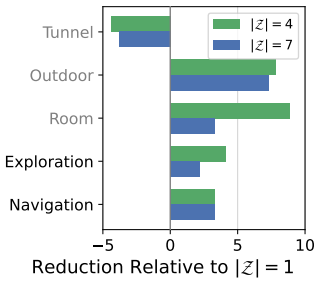

Figure 3: **Travel length reduction.** Comparison of 4 and 7 predictions vs. 1.

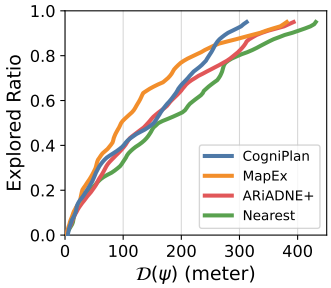

Figure 4: **Exploration progress in floor plans.** Average exploration rate vs. travel distance.

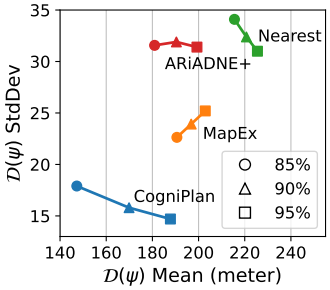

Figure 5: **Robustness to random starts.** Travel length at different exploration rates.

ploration, outperforming other methods beyond 90% coverage. We believe this may be because the predicted layouts, though coarse, provide informative structural priors that enable long-horizon planning and reduce redundant local actions, which in turn enables the near-linear exploration progress shown in Fig. 4. This is further supported by CogniPlan outperforming ARiADNE+, which does not explicitly predict environment layouts.

CogniPlan also demonstrates robustness to starting position variability in exploration. We evaluate performance on two similarly sized maps, each with five diverse starting points selected from different spatial regions. We report the mean and standard deviation of total travel distance at 85%, 90%, and 95% exploration coverage, which reflect near-complete exploration performance. As shown in Fig. 5, CogniPlan achieves the lowest average travel distance and standard deviation, indicating reduced sensitivity to starting position and more globally consistent planning (the closer to the bottom left corner, the better). At 90% and 95% coverage, it reduces standard deviation of $D(\psi)$ by over 50% compared to nearest and ARiADNE+. We attribute this improvement to the predictions providing global structural priors that guide the planner, reducing backtracking and redundant revisits.

4.3 Evaluation on Hardware

We validate CogniPlan through real-world exploration experiments conducted in a 30m $\times$ 10m indoor laboratory filled with cluttered objects such as tables, chairs, equipment, and moving people. Our mobile platform is a four-wheeled omnidirectional robot with a maximum speed set to 0.75 m/s, equipped with a Livox Mid-360 3D LiDAR for odometry and mapping. We use an Octomap resolution of 0.2m per cell and a waypoint node resolution of 0.8m. As observed, the robot generally completes exploration within 6 minutes, with a travel length of approximately 100 meters. Fig. 6 illustrates how the robot predicts layouts during exploration. We believe these results show the promise of CogniPlan's framework for practical use.

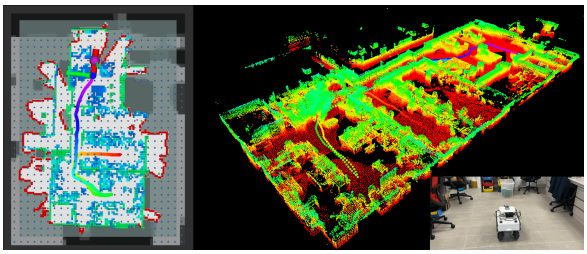

Figure 6: **Real-world exploration experiment in a 30m $\times$ 10m indoor laboratory.** We show our mobile robot, the lab environment, the intermediate Octomap, and the final point cloud with the robot's trajectory.

5 Conclusion

In this work, we proposed CogniPlan, a path planning framework that combines uncertainty from multiple explicit layout predictions with a graph attention-based planner trained via deep reinforcement learning. We demonstrate their synergistic effect, where the inpainting model amplifies non-trivial map details into uncertainty-aware graph representations to guide downstream planning. We show CogniPlan plan more efficient paths for both exploration and navigation tasks across simulated environments and realistic floor plans, all without additional training. We further deploy it on a mobile robot, showcasing its ability to handle real-world scenarios.

Limitations and Future Work

Generalization and Prediction Diversity. Although CogniPlan's planner network leverages graph attention to scale across arbitrary graph topologies, the inpainting model still requires a fixed-size occupancy map input. This imposes certain prior assumptions on the environment, which may limit the scalability and generalization of the overall framework, and leave a domain gap upon deployment. A possible solution is to adopt resolution-agnostic network architectures, coupled with more diverse training data, to enhance spatial understanding across varied environments.

One-Way Pipeline. After pre-training our inpainting model, the generator is frozen during planner training. That is, the planner must adapt to whatever predictions it receives, which limits continual improvement and hampers adaptation to novel layouts discovered online. To address this, we plan to explore end-to-end joint training, allowing planner gradients to flow into the generator, thereby improving both components simultaneously, although this approach requires careful architectural design. We also plan to use active sampling to select conditioning vectors in latent space, instead of relying on a fixed or handcrafted set.

Replanning Computation Overhead. We employ a lightweight inpainting model capable of rapidly inferring the layout; however, the main bottleneck lies in the graph rescheduling, where each new prediction triggers a complete rebuild of the robot's belief graph. This process can be accelerated by comparing successive predictions and incrementally updating the graph through selective addition and pruning.

A Validation in Gazebo Simulation

Environment Setup We validate CogniPlan's real-world performance in a high-fidelity Gazebo environment provided by Cao et al. [18, 19], using a four-wheel-drive mobile robot equipped with a 16-line Velodyne LiDAR for odometry and mapping. The environment incorporates realistic motion constraints, while a local planner ensures safe execution of waypoints generated by the high-level planner. During validation, we adopt the graph rarefaction strategy from [7] to reduce graph complexity. Specifically, we preserve a dense graph structure in predicted regions while applying rarefaction in known areas, and remove nodes that are not connected via a valid path to the robot's current node.

Medium-Scale Environment. We first evaluate CogniPlan in a $54\mathrm{m} \times 34\mathrm{m}$ indoor environment crafted by Long et al. [27], which contains dense clutter and obstacles. They also introduced HPHS, a hierarchical planner that directly samples from LiDAR data. We compare CogniPlan against the baselines tested and reported in [27], including a frontier-based approach [53] and an improved RRT planner TDLE [54]. The results are shown in Table 3, where we observe that CogniPlan produces shorter paths with lower variance, indicating more consistent and stable trajectories due to effective layout/uncertainty prediction. The corresponding trajectory is visualized in Fig. 7.

Table 3: **Comparison over 10 runs in the medium-scale environment.** We report the mean and standard deviation of travel distance and path efficiency relative to the explored area.

Method	Frontier-based	TDLE	TARE	HPHS	CogniPlan
$D(\psi)$ (m)	270.6($\pm$42.1)	290.5($\pm$35.6)	190.3($\pm$31.4)	176.9($\pm$24.7)	**170.9**($\pm$18.1)
Efficiency (m^2/m)	3.14	2.93	4.31	4.78	**4.95**

Large-Scale Environment. We further evaluate our method in a $130\mathrm{m} \times 100\mathrm{m}$ indoor setting comprising long, narrow corridors interconnected with spacious lobby areas. We compare CogniPlan against DSV Planner [26] and TARE [18], both of which represent state-of-the-art conventional planner for large-scale environments. We run each method five times and present the exploration progress in Fig. 8 and trajectory example in Fig. 7. We notice that CogniPlan consistently outperforms TARE and DSVP throughout the exploration process, with notably lower variance, highlighting that layout predictions can enhance both performance and robustness. Figure 9 visualizes the lay-

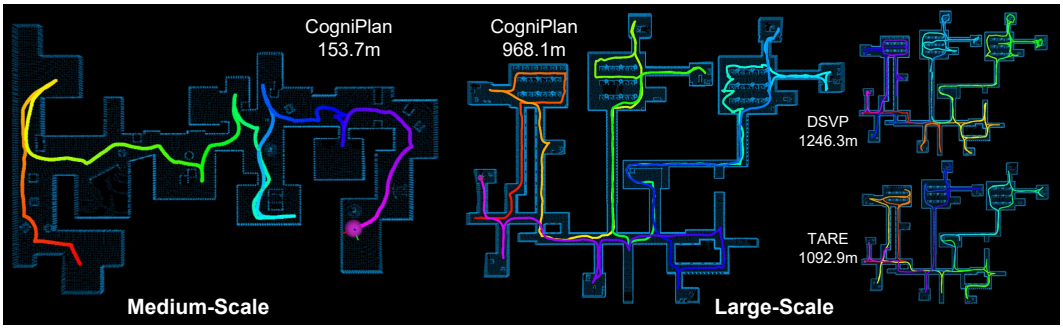

Figure 7: **Trajectories of CogniPlan and baseline planners in medium- and large-scale environment.** Colored lines represent the robot's motion trajectories, with the red-to-purple spectrum indicating progression from start to end.

out predictions during exploration. Despite the environment being significantly out-of-distribution from the training data of both the inpainting and planner models, the predicted layouts still effectively support planning and contribute to improved performance.

We believe this demonstrates the potential of the CogniPlan framework as a fully learning-based planning approach that enhances performance, path robustness, and explainability. It also opens up promising directions for future research in multi-agent planning and visual navigation.

Table 4: **Comparison over 5 runs in the large-scale environment.** We report the mean and standard deviation of travel distance and path efficiency relative to the explored volume.

Method	DSVP	TARE	CogniPlan
$D(\psi)$ (m)	1300.6($\pm$58.7)	1144.9($\pm$70.7)	**999.1**($\pm$17.1)
Efficiency (m^3/m)	4.03($\pm$0.19)	4.58($\pm$0.30)	**5.25**($\pm$0.09)

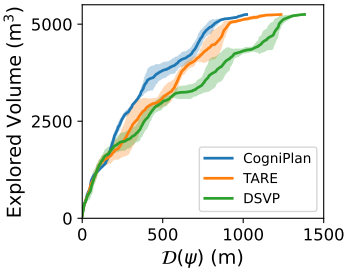

Figure 8: **Exploration progress in a large-scale environment.**

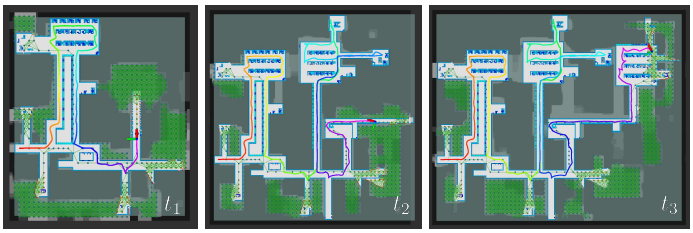

Figure 9: **Layout predictions at three different timesteps during exploration in a large-scale environment.** We also show the corresponding occupancy map (Octomap) and the rarefied graph (nodes and green edges).

B Framework Design Choices

Environment Types. *Room* consists of spaces with varying sizes and connectivity. *Tunnel* comprises long corridors, where the effects of each planning decision are significantly delayed. *Outdoor* is spacious with few obstacles or walls (see Fig. 10). We identify these three layout types as distinct and use them to train our inpainting network accordingly. Thus, we preserve these characteristics during planner evaluation as well.

Reconstruction Loss. The reconstruction term comprises L1 (pixel-level), Dice (region-level), and an additional L1 loss that is spatially discounted around the known areas. The L1 and Dice losses are complementary: L1 alone stabilizes gradients but often causes artifacts, whereas adding a small amount of Dice loss reduces these artifacts and promotes spatial connectivity in the predictions.

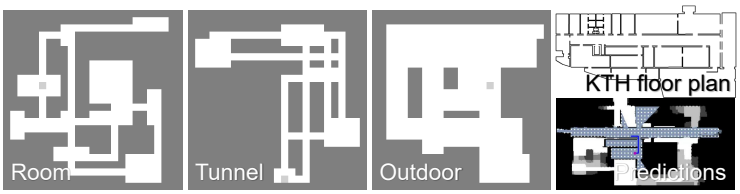

Figure 10: Examples of each environment type, along with an out-of-distribution prediction on the KTH floor plan.

Conditioning Vector. We explored various conditioning strategies, including VAE sampling and Monte Carlo dropout for probabilistic approximation, and found that one-hot and soft one-hot vectors yield the best diversity and plausibility. Conditioning vectors that are not valid probability distributions tend to produce unrealistic layouts (nearly all free space or all obstacles). The vector choices for $|\mathcal{Z}| = 7$ are illustrated in Fig. 2. We uniformly average predictions without weighting (Eq. 3), as diversity naturally decreases as the robot explores/observes more of the environment. That is, early predictions primarily rely on layout conditioning vectors, while later ones converge based on the robot's belief, forming a sense of conditional posterior.

$$\hat{\mathcal{M}}_{\text{all}} = \frac{1}{|\mathcal{Z}|} \sum_{z_i \in \mathcal{Z}} \text{Gen}\left(\mathcal{M} \mid z_i\right) \tag{3}$$

Reward functions. We address both exploration and navigation by leveraging conditional layout predictions to guide the graph attention network backbone. The two tasks share the same planner framework and both aim to complete the task with minimal travel length or makespan, but they differ in their input node features and in parts of the reward functions used to train the planner. Following [7] for exploration and [9] for navigation, both tasks reward task completion and penalize travel distance. Exploration further rewards the discovery of new frontiers, while navigation rewards proximity to the goal, providing denser feedback that facilitates deep RL training.

Acknowledgments

This work was partly supported by TL@NUS.

References

[1] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, pages 146–151. IEEE, 1997.

[2] H. H. González-Banos and J.-C. Latombe. Navigation strategies for exploring indoor environments. *The International Journal of Robotics Research*, 21(10-11):829–848, 2002.

[3] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart. Receding horizon" next-best-view" planner for 3d exploration. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1462–1468. IEEE, 2016.

[4] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011.

[5] D. Zhu, T. Li, D. Ho, C. Wang, and M. Q.-H. Meng. Deep reinforcement learning supervised autonomous exploration in office environments. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 7548–7555. IEEE, 2018.

[6] H. Li, Q. Zhang, and D. Zhao. Deep reinforcement learning-based automatic exploration for navigation in unknown environment. *IEEE transactions on neural networks and learning systems*, 31(6):2064–2076, 2019.

[7] Y. Cao, R. Zhao, Y. Wang, B. Xiang, and G. Sartoretti. Deep reinforcement learning-based large-scale robot exploration. *IEEE Robotics and Automation Letters*, 2024.

[8] L. Tai, G. Paolo, and M. Liu. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 31–36. IEEE, 2017.

[9] J. Liang, Z. Wang, Y. Cao, J. Chiun, M. Zhang, and G. A. Sartoretti. Context-aware deep reinforcement learning for autonomous robotic navigation in unknown area. In *Conference on Robot Learning*, pages 1425–1436. PMLR, 2023.

[10] R. Shrestha, F.-P. Tian, W. Feng, P. Tan, and R. Vaughan. Learned map prediction for enhanced mobile robot exploration. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1197–1204. IEEE, 2019.

[11] M. Luperto, L. Fochetta, and F. Amigoni. Exploration of indoor environments through predicting the layout of partially observed rooms. In *Proceedings of the 20th international conference on autonomous agents and multiagent systems*, pages 836–843, 2021.

[12] G. Georgakis, B. Bucher, A. Arapin, K. Schmeckpeper, N. Matni, and K. Daniilidis. Uncertainty-driven planner for exploration and navigation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 11295–11302. IEEE, 2022.

[13] V. D. Sharma, A. Singh, and P. Tokekar. Pre-trained masked image model for mobile robot navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5126–5133. IEEE, 2024.

[14] C. Ho, S. Kim, B. Moon, A. Parandekar, N. Harutyunyan, C. Wang, K. Sycara, G. Best, and S. Scherer. Mapex: Indoor structure exploration with probabilistic information gain from global map predictions. In *2025 International Conference on Robotics and Automation (ICRA)*, 2025.

[15] R. A. Epstein, E. Z. Patai, J. B. Julian, and H. J. Spiers. The cognitive map in humans: spatial navigation and beyond. *Nature neuroscience*, 20(11):1504–1513, 2017.

[16] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[17] Y. Cao, Y. Wang, A. Vashisth, H. Fan, and G. Sartoretti. Context-aware attention-based network for informative path planning. In *6th Annual Conference on Robot Learning*, 2022.

[18] C. Cao, H. Zhu, H. Choset, and J. Zhang. Tare: A hierarchical framework for efficiently exploring complex 3d environments. In *Robotics: Science and Systems*, volume 5, page 2, 2021.

[19] C. Cao, H. Zhu, Z. Ren, H. Choset, and J. Zhang. Representation granularity enables time-efficient autonomous exploration in large, complex worlds. *Science Robotics*, 8(80):eadf0970, 2023.

[20] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[21] D. Holz, N. Basilico, F. Amigoni, and S. Behnke. Evaluating the efficiency of frontier-based exploration strategies. In *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, pages 1–8. VDE, 2010.

[22] M. Kulich, J. Faigl, and L. Přeučil. On distance utility in the exploration task. In *2011 ieee international conference on robotics and automation*, pages 4455–4460. IEEE, 2011.

[23] L. Lu, C. Redondo, and P. Campoy. Optimal frontier-based autonomous exploration in unconstructed environment using rgb-d sensor. *Sensors*, 20(22):6507, 2020.

[24] A. Visser, M. Van Ittersum, L. A. González Jaime, and L. A. Stancu. Beyond frontier exploration. In *RoboCup 2007: Robot Soccer World Cup XI 11*, pages 113–123. Springer, 2008.

[25] C. Connolly. The determination of next best views. In *Proceedings. 1985 IEEE international conference on robotics and automation*, volume 2, pages 432–435. IEEE, 1985.

[26] H. Zhu, C. Cao, Y. Xia, S. Scherer, J. Zhang, and W. Wang. Dsvp: Dual-stage viewpoint planner for rapid exploration by dynamic expansion. In *2021 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 7623–7630. IEEE, 2021.

[27] S. Long, Y. Li, C. Wu, B. Xu, and W. Fan. Hphs: Hierarchical planning based on hybrid frontier sampling for unknown environments exploration. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12056–12063. IEEE, 2024.

[28] B. Zhou, Y. Zhang, X. Chen, and S. Shen. Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning. *IEEE Robotics and Automation Letters*, 6(2):779–786, 2021.

[29] A. Kamalova, S. G. Lee, and S. H. Kwon. Occupancy reward-driven exploration with deep reinforcement learning for mobile robot system. *Applied Sciences*, 12(18):9249, 2022.

[30] Y. Tao, E. Iceland, B. Li, E. Zwecher, U. Heinemann, A. Cohen, A. Avni, O. Gal, A. Barel, and V. Kumar. Learning to explore indoor environments using autonomous micro aerial vehicles. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15758–15764. IEEE, 2024.

[31] F. Chen, J. D. Martin, Y. Huang, J. Wang, and B. Englot. Autonomous exploration under uncertainty via deep reinforcement learning on graphs. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6140–6147. IEEE, 2020.

[32] M. Likhachev, D. I. Ferguson, G. J. Gordon, A. Stentz, and S. Thrun. Anytime dynamic a*: An anytime, replanning algorithm. In *Proceedings of the 15th International Conference on Automated Planning and Scheduling (ICAPS)*, volume 5, pages 262–271, 2005.

[33] S. Koenig and M. Likhachev. D* lite. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI/IAAI)*, volume 15, pages 476–483, 2002.

[34] J. J. Kuffner and S. M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, pages 995–1001. IEEE, 2000.

[35] 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)*, pages 3067–3074. IEEE, 2015.

[36] J. Zhang, J. T. Springenberg, J. Boedecker, and W. Burgard. Deep reinforcement learning with successor features for navigation across similar environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2371–2378. IEEE, 2017.

[37] M. Everett, Y. F. Chen, and J. P. How. Motion planning among dynamic, decision-making agents with deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3052–3059. IEEE, 2018.

[38] J. Liang, Y. Cao, Y. Ma, H. Zhao, and G. Sartoretti. Hdplanner: Advancing autonomous deployments in unknown environments through hierarchical decision networks. *IEEE Robotics and Automation Letters*, 2024.

[39] M. Wei, D. Lee, V. Isler, and D. Lee. Occupancy map inpainting for online robot navigation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8551–8557. IEEE, 2021.

[40] L. Ericson and P. Jensfelt. Beyond the frontier: Predicting unseen walls from occupancy grids by learning from floor plans. *IEEE Robotics and Automation Letters*, 2024.

[41] S. Baek, B. Moon, S. Kim, M. Cao, C. Ho, S. Scherer, and J. h. Jeon. Pipe planner: Path-wise information gain with map predictions for indoor robot exploration. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

[42] K. Katyal, K. Popek, C. Paxton, P. Burlina, and G. D. Hager. Uncertainty-aware occupancy map prediction using generative networks for robot navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5453–5459. IEEE, 2019.

[43] S. K. Ramakrishnan, Z. Al-Halah, and K. Grauman. Occupancy anticipation for efficient exploration and navigation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 400–418. Springer, 2020.

[44] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022.

[45] D. P. Kingma, M. Welling, et al. Auto-encoding variational bayes, 2013.

[46] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[47] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.

[48] D. M. S. Tan, Y. Ma, J. Liang, Y. C. Chng, Y. Cao, and G. Sartoretti. Ir 2: Implicit rendezvous for robotic exploration teams under sparse intermittent connectivity. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13245–13252. IEEE, 2024.

[49] Y. Cao, Z. Sun, and G. Sartoretti. Dan: Decentralized attention-based neural network for the minmax multiple traveling salesman problem. In *International Symposium on Distributed Autonomous Robotic Systems*, pages 202–215. Springer, 2022.

[50] F. Chen, S. Bai, T. Shan, and B. Englot. Self-learning exploration and mapping for mobile robots via deep reinforcement learning. In *Aiaa scitech 2019 forum*, page 0396, 2019.

[51] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

[52] A. Aydemir, P. Jensfelt, and J. Folkesson. What can we learn from 38,000 rooms? reasoning about unexplored space in indoor environments. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4675–4682. IEEE, 2012.

[53] J. Oršulić, D. Miklić, and Z. Kovačić. Efficient dense frontier detection for 2-d graph slam based on occupancy grid submaps. *IEEE Robotics and Automation Letters*, 4(4):3569–3576, 2019.

[54] X. Zhao, C. Yu, E. Xu, and Y. Liu. Tdle: 2-d lidar exploration with hierarchical planning using regional division. In *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, pages 1–6. IEEE, 2023.