Uncertainty-Aware Map-Space Dynamics Models for Manipulation-Enhanced Mapping

Nils Dengler[2*] Joao Marcos Correia Marques[1*] Tobias Zaenker[2] Vamsi Kalagaturu[2]
Shenlong Wang[1] Maren Bennewitz[2] Kris Hauser[1]

I. INTRODUCTION

A critical task in many robotic applications is acquiring and consistently updating an accurate and detailed model of the environment to plan and execute diverse actions in it. This is especially true in interactive scenes, where objects can be moved by the robot or humans. To efficiently maintain a map representation of such environments, one solution is to apply Next Best Viewpoint planning (NBV) [11] to reduce the uncertainty about the environment while minimizing the required number of observations to update the map. However, in confined and cluttered scenes, e.g. shelves, observing all objects in the scene is not always possible due to occlusions, leading to an incomplete representation and, consequently, difficulties in searching and retrieving desired objects.

We propose a modular and extensible policy inspired by Partially Observable Markov Decision Processes (POMDP) that leverages learned environment dynamics in map space and uncertainty-aware map completion to efficiently explore environments with movable objects. Our approach computes the action sequence that, in expectation, maximizes the agent's information gain over a finite horizon, as depicted in Figure 1. As the central objective in manipulation-enhanced mapping is to minimize map entropy, model overconfidence in either dynamics prediction or map completion prediction would result in incomplete exploration with improper early termination. Therefore, we achieve confidence-calibrated map and push prediction by employing evidential deep learning [8] in our map completion and push prediction models. We experimentally show that our pipeline overcomes prior work [4] and strong baselines in terms of final map metric-semantic accuracy and confidence calibration.

II. METHODS

A. Problem Definition

In this work, we consider a confined environment with movable objects of varying sizes and orientations. However, some objects may be unobservable from any viewpoint due to occlusions by others. A robotic arm, equipped with a wrist-mounted RGB-D camera and a gripper, aims to build an accurate map of the current workspace configuration C_W[1] after a sequence of actions, which can be either taking an RGB-D image or performing a manipulation (e.g., a push) to change object configurations and reveal occluded areas.

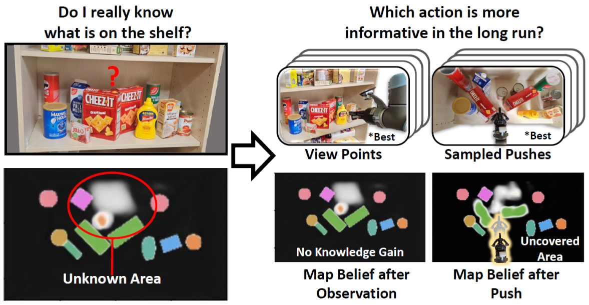

Fig. 1: Given a partial map of the environment, the robot has to decide whether an observation or a manipulation action at the current state best reduces the map uncertainty in the long run.

Let Φ^t represent the robot's internal environment map at time t. When manipulating the environment, it causes a transition on the workspace configuration space from $c_w^t \mapsto c_w^{t+1} \in C_W$ according to the environment's dynamics. Further, whenever the robot chooses to take another RGB-D observation, it updates its internal environment representation according to its belief update, $\Phi^t \rightarrow \Phi^{t+1}$.

The problem, considered in this work, is to determine the most informative sequence of actions for a robot, within a given action budget, that minimizes the difference between the robot's internal map belief and the true environment configuration, using a similarity metric, like IoU.

B. Overview

We model this problem as a Partially Observable Markov Decision Process (POMDP) with the parameters S, A, T, R, Z, O. The state S consists of the fully observable robot configuration and the partially observable workspace configuration, detected through RGB-D and semantic observations. The transition function $T : C_w \mapsto C_w$ is initially unknown, and the workspace is represented by a dense voxel map with height H, width W, depth D, and M classes, defining the state as $q \in \mathbb{R}^{dof} \cup s \in \mathbb{N}^{HxWxD}$.

The action space A includes two types of actions: **observation actions**, which change only the robot's state and produce partial observations $o \in O$ via the observation function Z, and **interactive actions**, which alter the workspace configuration without generating new observations beyond proprioception. The reward function R is the negative mean voxel-wise cross-entropy between Φ^t (the robot's internal map) and c_w^t (the true configuration).

To solve the POMDP, the agent must update its belief about the map state after both manipulation and observation actions. However, due to the high dimensionality of the map,

$*$ These authors contributed equally to this work.

1. University of Illinois at Urbana-Champaign, IL, USA.
2. Humanoid Robots Lab, University of Bonn, Germany

This work has been supported by the Lamarr Institute for Machine Learning and Artificial Intelligence, Germany.

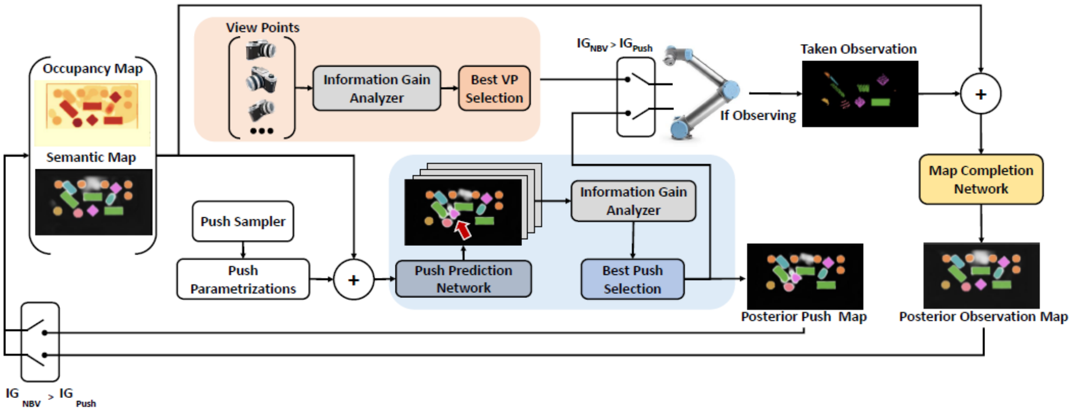

Fig. 2: Overview of our framework for viewpoint manipulation planning. From a prior map belief, our pipeline predicts the posterior and selects which action, i.e., observation or manipulation, is best to perform and brings the highest information gain in the long run.

traditional belief updates for POMDPs are impractical. A common approximation assumes map cells are independent, as in occupancy grid mapping [9], but this leads to incorrect environment representations. For example, two maps with the same number of occupied cells—one with salt-and-pepper noise and another with clustered voxels—are equally likely under a naive uniform prior, though the former is unrealistic since objects in the real world tend to be contiguous.

To address this, we propose using uncertainty-calibrated deep learning models to predict a factorized belief distribution. This approach better aligns the robot's factorized belief to more plausible map distribution than naive independent updates. For this we use network architectures similar to those described by Georgakis *et al.* [5], with the exception that the output heads are set to be posterior networks. Their losses, data augmentation and collection are described in more depth in the following sections.

C. Ideal Factorized Belief Update

We consider an RGB-D image with added semantics as observation at time t, $o_t \in \mathcal{O}$, taken from a camera fixed to the robot's end-effector. We assume, that the robot can sample images from a finite set of viewpoints $v_k \in V$. Following occupancy grid mapping literature [9], we represent our belief over the environment state s^τ as a dense HxWxD voxel grid of independent cells, $m_{i,j} \in \mathcal{M}$, each tracking their own occupancy probability and a HxWxM 2D semantic map of independent cells $s_{i,j}$.

We propose to retain the independent cell representation, but alter the cell probability updates to better align the implicit belief with a properly calculated one. We presume there exists a function $\Omega(\Phi^t, s^\tau) = P(s^\tau|\Phi^t)$ that outputs an accurate estimate of the probability of any state s_τ given the current independently factorized map representation at time t, with Φ^t and $z(o_t, s^\tau)$ as our chosen observation model. The POMDP belief update equations [6] are:

$$P^t(s^\tau) = \frac{1}{\eta} z(o_t, s^\tau) \sum_{s' \in S} T(s^\tau, a, s') \Omega(\Phi^{t-1}, s'), \quad (1)$$

where s' is any other possible state and $P^t(s^\tau)$ indicates the probability of state s^τ at time t, with η a normalizing constant. From those new probabilities, we derive the marginalized occupancy probability of any given element in the map. The same may be done for semantic mapping using a naive average update of the semantics as in [7].

D. Making the belief updates tractable

According to our problem definition, observation actions will never cause a transition on the non-observable part of the state. We propose estimating this update using a deep posterior network [10], $\upsilon_o(m_{t-1}, o_t)$, where o_t denotes the observation at time *t*, leveraging the neural network's averaging tendency to create an implicit Monte Carlo estimate of the map cell update.

1) Dataset Generation: To train this model, we collect the following dataset: each data point consists of $(m_{gt}, o_1, o_2, \cdots, o_n)$, where m_{gt} is the ground truth 3D metric-semantic voxel map of a given environment with randomly placed objects and $o_1, \cdots, o_n$ as the depth and semantic images seen from a set of n discrete pre-selected viewpoints in the environment.

2) Model Training: We train the model as follows: Every epoch, for every data point, we sample a sequence of l posed depth and semantic images, $o'_0, o'_1, \cdots, o'_l$, without replacement. Let $\tilde{m}_{ot} = \upsilon_{oo}(\tilde{m}_{t-1}, o'_t)$ be a Beta distribution for occupancy and $\tilde{m}_{st} = \upsilon_{os}(\tilde{m}_{t-1}, o'_t)$ a Dirichlet distribution for each semantic voxel. Let α_i be the α parameters and p_i be the mean of the Dirichlet (or Beta) distribution for a single map element *i* at time t and let $S_i = \sum \alpha_i$ be its concentration parameter. Finally, let $\tilde{\alpha}_i = y_i + (1 - y_i) \odot \alpha_i$, where $\odot$ is the element-wise multiplication and y_i its ground truth class. We use the evidential uncertainty-aware cross-entropy loss from Sensoy *et al.* [8] to train the network.

The total loss for training the metric-semantic belief update network is then the sum of the semantic and occupancy losses $L_i^o + L_i^s$ summed over l observations.

E. Handling the belief update after manipulation actions

Similarly, by assumption, the manipulation action itself does not generate any new observations. We thus learn this update through an action specific network, υ_a, with action specific parametrization.

1) Dataset Generation: We collect a dataset where each sample has the form $(m_{gt}^{pre}, m_{gt}^{post}, o_1, o_2, \cdots, o_n, \zeta)$, where m_{gt}^{post} are the ground truth maps pre and post manipulation, ζ is the action-dependent parametrization of the executed manipulation sampled according to Sec II-F and $o_1, \cdots, o_n$ are the images collected pre manipulation.

2) Action Belief Update Training: Every batch, we sample a sequence of $l \in [1, 10]$ images without replacement, and recursively obtain the beliefs from the map observation belief propagation network $, \upsilon_o$, at every time step t: $\tilde{m}_t = \upsilon_o(\tilde{m}_{t-1}, o_t)$ and obtain the output via $\upsilon_a(\tilde{m}_l, \zeta)$. υ_a has 3 outputs: The occupancy and semantic posteriors for the map after the push (like υ_o) and a beta distribution for the estimated map occupancy map difference, used as an auxiliary task for the network, whose ground truth, m_{gt}^{change} is derived from the difference between m_{gt}^{pre} and m_{gt}^{post}. As before, the network heads are trained using the uncertainty-aware cross-entropy loss [8]. The final loss is the sum of the occupancy, semantic and differences loss for the three outputs of $\upsilon_a(\tilde{m}_l, \zeta_i)$. Onwards, We refer to the factorized occupancy and semantic belief representations (maps) at time t by $\Phi^t = (\Phi_O^t, \Phi_S^t)$,respectively

F. Push Sampling

Our manipulation action of choice is pushing. To compute valid push candidates using Φ_O^t, we first compute the frontier points from the shelf entry and sample k of them uniformly at random as start points for the pushes. We test the start points of the k sampled pushes for collisions against high confidence voxels in Φ_O^t. For each valid start point, we sample a likely occupied point in Φ_O^t near it to obtain the push direction and sample a push distance uniformly at random between 50 and 150 mm. We then obtain a valid motion plan for the entire trajectory using a sampling based motion planner and use it to calculate an approximation of the robot's swept volume within the voxel map of interest and start and end points, which are the action parametrization inputs used in the υ_a.

G. Solving the POMDP

Volumetric Information Gain (VIG) [3] can be used with submodular optimization in static scenes to efficiently solve Next-Best-View planning and sensor placement. Therefore, a greedy policy for solving this problem would lead to bounded suboptimality. While VIG's submodularity does not hold in general for a dynamic scene representation, we assume that manipulation actions are sufficiently rare in occurrence, that the submodularity assumption is still valid for the dominant part of the policy. As such, we propose a 2-step greedy search policy to solve this POMDP. Let $v_i \in V$ be the possible views on the camera array V. Furthermore, let $\Theta_k^t \subseteq \Theta$ be a set of K sampled pushes from the sampling method at time t explained in Sec. II-F.

In our two-step greedy search, we only need to consider two possible kinds of action sequences: first, taking two observations and second, performing a manipulation action followed by an observation, namely (v_t, v_{t+1}) or (θ_t, v_{t+1}).

Let Γ be a volumetric Occlusion-aware Information Gain [3] calculation module. Let $IG(v_i, \cdots, v_k | \Phi^o) = \Gamma(v_i, \cdots, v_k, | \Phi_o^t)$ denote the OIG of the non-redundant rays from views $v_i, \cdots, v_k$ on the voxel grid Φ_o at time t. We can define the two most informative consecutive pushes (v_t^*, v_{t+1}^*) at time t as:

$$(v_t^*, v_{t+1}^*) = \underset{v_t, v_{t+1} \in V}{argmax}\, IG(v_t, v_{t+1} | \Phi_o^t) \tag{2}$$

Let $\tilde{\Phi}_{\theta_t}^{t+1} = \upsilon_a(\Phi^t, \theta_t)$ denote the predicted belief from the push prediction network when given action $\theta_t \in \Theta_k^t$ as input. We can define the most informative 1-step push, θ_t^* and its associated most informative view $v_{\theta_t}^*$,as:

$$\theta_t^*, v_{\theta_t}^* = \underset{\theta_t \in \Theta_k^t}{argmax}\ \underset{v_{t+1} \in V}{max}\, IG(v_{t+1} | \tilde{\Phi}_{\theta_t}^{t+1}) \tag{3}$$

Our agent then decides the action a_t to take according to:

$$a_t = \begin{cases} v_t^* & \text{if } \gamma IG(v_t^*, v_{t+1}^* | \Phi_o^t) > IG(v_{\theta_t}^* | \tilde{\Phi}_{\theta_t^*}^{t+1}) \\ \theta_t^* & \text{otherwise} \end{cases} \tag{4}$$

Where γ is a discount factor, that is set to 1.1 in this work, to account for the extra cost of manipulating the environment. If $a_t = \theta_t^*$, then $\Phi^{t+1} = \tilde{\Phi}_{\theta_t^*}^{t+1}$. If a_t is an observation action, we get the observation at time t, o_t, and use the map completion module to obtain the new belief $\Phi^{t+1} = \upsilon_o(\Phi^t, o^t)$. We then repeat this until the maximum number of actions has been performed, or a threshold for full map completion has been reached, which is set to 95%.

III. EXPERIMENTS

We perform two experiments to highlight our method's improvements in map completeness, accuracy and confidence calibration over a random agent, a series of ablations and the method proposed by Dengler *et al.* [4]. We also perform an experiment to examine the influence of evidential vs non-evidential map completion on simulated data.

A. Experimental Setup

For the experimental evaluation, we set up a shelf scene with a UR5 arm for observation and action execution in PyBullet [2]. The robot is equipped with a Robotiq parallel-jaw gripper and an RGB-D camera for observations.

To sample realistic object configurations, a total of 14 different object categories from the YCB dataset are used and sampled in a shelf board of size $(0.8 \times 0.4 \times 0.4)m$. We sample object configurations on the shelf following a stochastic method that considers class dependencies and efficient free space coverage for placing. This method allows for the sampling of varied object configurations, numbers and classes, including fully random scenes and structured scenes, such as those found in grocery store shelves. In addition, we implement a simple voxel-based GPU occupancy grid mapper for collecting ground truth data.

B. Baselines & Metrics

We re-implemented the approach by Dengler *et al.* [4] (RL VPP) using their provided weights, as their setup matched ours. Additionally, we created a Random baseline, which samples views randomly and uses standard metric-semantic

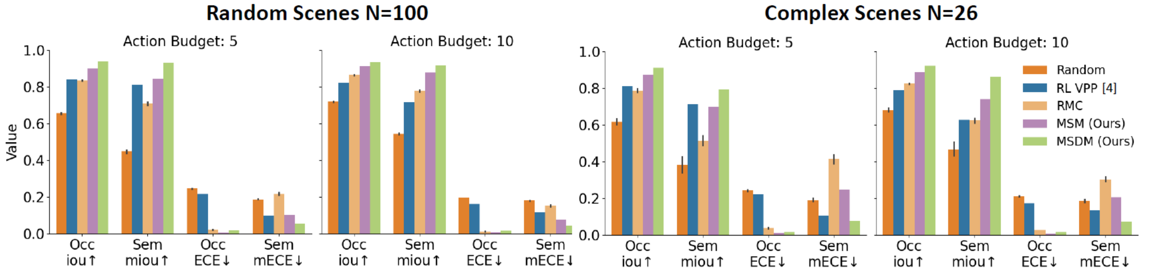

Fig. 3: Calibration and Accuracy results for different agents in both random and handcrafted challenging scenarios for exploration. Our results show that our map completion model is quite accurate and well-calibrated.

Model (budget)	Occ. IoU↑	Sem. mIoU↑	Occ. ECE ↓	Sem. mECE↓
Non-Evidential (5)	0.8570	0.6104	0.0370	0.3411
Evidential (5)	**0.8750**	**0.6968**	**0.0109**	**0.2470**
Non-Evidential (10)	**0.8927**	0.6899	0.0257	0.2683
Evidential (10)	0.8859	**0.7423**	**0.0074**	**0.2062**

TABLE I: Comparison of OBM performances of the Information Gain + Map completion agent using both evidential and non-evidential completion models.

occupancy mapping [9]. For ablation, we combined the Random agent with our map completion network (RMC) to show the advantage of our view selection method. Finally, we evaluate our pipeline with manipulation (using Map Space Dynamics Models, MSDM) and without (MSM) to demonstrate that our agent reveals more of a scene by effectively using pushing.

We compare their confidence calibration using expected calibration error (mECE) and their segmentation performance with mIoU with the ground truth.

C. View point Manipulation Mapping Comparisons

We evaluated our method on 100 randomly sampled scenes and 26 handcrafted scenes that require pushing to fully reveal their contents. All agents start with a naive prior and aim to explore the environment with limited action budgets of 5 (short) and 10 (long) steps. For stochastic methods like Random and RMC, we repeated each scenario four times, reporting the mean and standard deviation.

Fig. 3 shows that our method outperforms all baselines in both map accuracy and calibration across both complexity levels and budgets. While all methods improve with more steps, our method achieves over 90% IoU after just five steps, with minimal gains from additional actions. The higher IoU compared to MSM highlights the benefit of manipulation actions (pushes) in increasing map knowledge while maintaining low ECE. The RMC ablation using our map completion network shows significant accuracy improvement over the Random baseline but is still outperformed by our full pipeline with viewpoint selection.

D. Influence Of Evidential Networks on Task Performance

To evaluate whether evidential training improves performance in the given task, we train a map completion model with a near-identical architecture to our map prediction model, but using traditional cross-entropy loss instead of evidential learning. Both models were tested on our MSM pipeline, and their results are shown in Table I.

As expected, the agent using the evidential map completion model achieved significantly higher occupancy and semantic IoUs with a smaller budget and maintained an advantage in the more challenging semantic task with a larger budget. Additionally, its predictions were better calibrated in both semantics and occupancy across all budgets. This indicates that evidential deep learning provides better-calibrated and more informative map completions for OBM agents and potentially other deep-learning-based active perception tasks.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a POMDP-inspired policy solver, that decides between different action types to generate an uncertainty-aware map-apace dynamics model as belief. In contrast to prior work [4], our pipeline does not switch between modular action types, e.g., observing and manipulating, according to a simplistic heuristic, but considers all action types to be equally effective and decides according to the best informative outcome. Our results show the improved performance of our system in comparison to baselines and ablations in terms of occupancy and semantics map accuracy and demonstrate that our agent is able to reason about map dynamics and impact of actions to the scene.

REFERENCES

[1] J Chase Kew *et al.*, “Neural Collision Clearance Estimator for Batched Motion Planning,” in *Algorithmic Foundations of Robotics XIV*, S. M. LaValle *et al.*, Eds., Cham: Springer International Publishing, 2021, pp. 73–89.

[2] E. Coumans and Y. Bai, *Pybullet, a python module for physics simulation for games, robotics and machine learning*, http://pybullet.org, 2016–2021.

[3] J. Delmerico *et al.*, “A comparison of volumetric information gain metrics for active 3D object reconstruction,” *Autonomous Robots*, vol. 42, no. 2, pp. 197–208, 2018.

[4] N. Dengler *et al.*, *Viewpoint Push Planning for Mapping of Unknown Confined Spaces*, 2023.

[5] G. Georgakis *et al.*, “Learning to Map for Active Semantic Goal Navigation,” in *International Conference on Learning Representations*, 2022.

[6] L. P. Kaelbling *et al.*, “Planning and acting in partially observable stochastic domains,” *Artificial Intelligence*, vol. 101, no. 1, pp. 99–134, 1998.

[7] J. M. C. Marques *et al.*, *On the Overconfidence Problem in Semantic 3D Mapping*, 2023.

[8] M. Sensoy *et al.*, “Evidential Deep Learning to Quantify Classification Uncertainty,” in *Advances in Neural Information Processing Systems*, S Bengio *et al.*, Eds., vol. 31, Curran Associates, Inc., 2018.

[9] S. Thrun *et al.*, “Probabilistic robotics. 2005,” *Massachusetts Institute of Technology, USA*, 2005.

[10] D. Ulmer *et al.*, “Prior and Posterior Networks: A Survey on Evidential Deep Learning Methods For Uncertainty Estimation,” *PMLR*, 2023.

[11] R. Zeng *et al.*, “View planning in robot active vision: A survey of systems, algorithms, and applications,” *Computational Visual Media*, 2020.