Map Space Belief Prediction for Manipulation-Enhanced Mapping

Joao Marcos Correia Marques[1*] Nils Dengler[2,3,4*] Tobias Zaenker[2,4] Jesper Mücke[2]
Shenlong Wang[1] Maren Bennewitz[2,3,4] Kris Hauser[1]
* These authors contributed equally to this work
1. University of Illinois at Urbana-Champaign, IL, USA 2. Humanoid Robots Lab, University of Bonn, Germany
3. The Lamarr Institute, Bonn, Germany 4. The Center for Robotics, University of Bonn, Germany

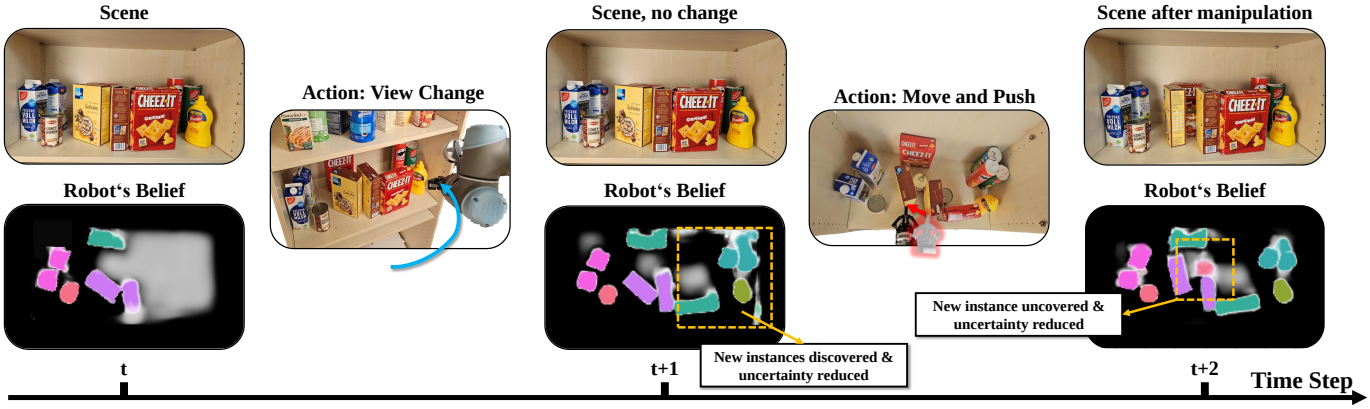


Fig. 1: Example scenario with occlusions in a confined shelf environment. Given a current partial map of the environment (belief t), our planner decides whether gathering another observation or manipulating the scene would be best to reduce map uncertainty. In this example, first a observation action would increase environmental knowledge, followed by a push to unveil the hidden can behind the two boxes at time $t+2$. The predicted belief map is visualized as a top-down projection of the shelf, ignoring the occluding top shelf board.

***Abstract*—Searching for objects in cluttered environments requires selecting efficient viewpoints and manipulation actions to remove occlusions and reduce uncertainty in object locations, shapes, and categories. In this work, we address the problem of manipulation-enhanced semantic mapping, where a robot has to efficiently identify all objects in a cluttered shelf. Although Partially Observable Markov Decision Processes (POMDPs) are standard for decision-making under uncertainty, representing unstructured interactive worlds remains challenging in this formalism. To tackle this, we define a POMDP whose belief is summarized by a metric-semantic grid map and propose a novel framework that uses neural networks to perform map-space belief updates to reason efficiently and simultaneously about object geometries, locations, categories, occlusions, and manipulation physics. Further, to enable accurate information gain analysis, the learned belief updates should maintain calibrated estimates of uncertainty. Therefore, we propose Calibrated Neural-Accelerated Belief Updates (CNABUs) to learn a belief propagation model that generalizes to novel scenarios and provides confidence-calibrated predictions for unknown areas. Our experiments show that our novel POMDP planner improves map completeness and accuracy over existing methods in challenging simulations and successfully transfers to real-world cluttered shelves in zero-shot fashion.**

I. Introduction

Active sensing has long been studied in robotics for tasks such as exploring an unknown environment [1], complete 3D object model acquisition [22], and searching for an unobserved target object [26, 12, 47]. To build complete maps as efficiently as possible, Next Best View (NBV) planning [46] is often employed to reduce the uncertainty about the map as quickly as possible. Although NBV planning offers an approach for static scenes in which the robot simply moves the camera passively through free space, there are many applications, such as household and warehouse robotics, in which robots may need to manipulate the environment in order to gain better viewpoints [11, 33]. We refer to this problem as *Manipulation-Enhanced Mapping* (MEM). MEM offers two significant new challenges beyond standard NBV problems. First, in order to decide when and where to manipulate objects, the robot should reason about how object movement may affect previously occluded regions. Second, it must anticipate the impact of manipulations on observed objects and possibly partially-observed or unobserved objects. For example, pushing boxes in a grocery shelf backward will move all objects simultaneously until the furthest, occluded one, hits a wall.

MEM is related to the *mechanical search* problem [9] in which the robot manipulates clutter to reveal a target object. Prior approaches in mechanical search tend to rely on restrictive assumptions, such as a static viewpoint, which ignores a robot's ability to look around obstacles [18, 39]. Other studies assume full observability of object dynamics and poses [37] or are limited to a fixed set of predefined objects [44]. These assumptions are too limiting for complex cluttered scenes like shelves. The most closely-related work to ours is Dengler et al. [11], who address these limitations by training a reinforcement learning policy for viewpoint planning

and learn a push scoring network from human annotations to derive manipulation actions, switching between manipulation and manipulation when the information gain from obtaining novel images of the environment saturates. However, their approach to switching between action modes is inefficient, since waiting for information gain saturation to perform a push results in the agent sampling the environment several times to reveal details that could have been revealed easily through manipulation. Furthermore, the proposed method does not update its environmental map after a push, leaving the viewpoint planner, conditioned solely on this outdated map, less capable of exploiting the newly revealed space.

This paper formulates the MEM problem as a Partially Observable Markov Decision Process (POMDP) in the belief space of semantic maps. By maintaining map-space beliefs, our approach is applicable to unstructured cluttered environments with an arbitrary numbers of objects. The POMDP computes the next best viewpoint or manipulation action that maximizes the agent's expected information gain over a short horizon (Fig. 1) in a receding-horizon fashion. Our approach leverages neural network methods for map-space belief propagation, which have been shown in the object goal navigation literature to drastically improve map completion rates and offer better guidance for object search [12, 47]. The key challenge in belief propagation with manipulation actions is that they often reduce certainty when the object's dynamics are unknown or the robot interacts with unobserved objects. To address this challenge, we introduce the Calibrated Neural-Accelerated Belief Update (CNABU) technique to learn belief propagation models for both observation (obtaining new images from the environment) and manipulation actions. Confidence calibration is especially important for belief propagation because overconfidence in either object dynamics or map prediction can result in ineffective exploration and/or early termination. We employ evidential deep learning to obtain better off-the-shelf model calibration [38].

Our experiments in simulation environments demonstrate that our proposed MEM planner outperforms prior work [11] and CNABU-enhanced baselines in terms of metric-semantic accuracy. Furthermore, we perform hardware experiments with a UR5 robot equipped with a gripper and an in-hand camera, demonstrating zero-shot transfer of the learned models, and showing the efficacy of our method in mapping of cluttered shelves. An implementation of our method can be found on Github[1].

II. Related Work

A. Next Best Viewpoint Planning

NBV planning is a well-researched approach in the area of active vision that has been applied to both object reconstruction and large-scale scene mapping. Generally, NBV consists of two steps: First sampling view candidates, then evaluating which candidate is the best. For object reconstruction tasks like [17], views are usually sampled from a fixed set around the object. For large-scale scenes, sampling is more challenging. Monica and Aleotti [29] sample at the contour of the explored scene. Other approaches sample at either predefined or dynamically detected regions of interest. For the evaluation, most approaches compute an estimated information gain to determine the utility of a view. The information gain is often based on the expected entropy reduction, e.g. by counting unknown voxels in the field of view. Other approaches like Hepp et al. [16] rely on a learned utility to predict the best view. In this work, we build upon existing concepts of NBV planing, but enhance them by incorporating manipulation actions to interactively shape and explore the environment, allowing the robot to gather richer information and adapt its strategy based on both observation and interaction.

B. Mechanical Search in Shelves and Piles

Mechanical search algorithms [9, 18, 39] locate and extract one or multiple target objects from a given scene, while dealing with confined spaces, occlusion and object occurence correlations. The task consists of multiple steps, i.e., visual reasoning, motion and action planning as well as their precise execution. For visual reasoning, current research demonstrates that the scene can be effectively explored by interacting with objects [3, 11, 26, 33] to actively reduce or overcome occlusions, but most works consider a fixed viewpoint [18, 9, 39].

Kim et al. [21] propose a method for locating and retrieving a target object using both pushing and pick-and-place actions. However, their approach relies on a fixed camera, lacks a long-term map, and rebuilds environmental knowledge from scratch with each observation. Therefore, the approach can lead to unnecessary manipulation actions, as the target may already be visible from other viewpoints. In the context of planning for Manipulation Among Movable Objects (MAMO) [36, 40], Saxena and Likhachev [37] introduced a method for object retrieval in cluttered, confined spaces. Despite achieving strong retrieval performance, their approach depends entirely on prior knowledge of object shapes and dynamics. Pajarinen et al. [31] propose a POMDP formulation for manipulating objects under uncertain segmentation using a particle belief representation, but limit their analysis to fixed viewpoints and prehensile manipulation to enable efficient belief propagation. Muguira-Iturralde et al. [30] propose a planner based on two-level hierarchical search to enable visibility-aware navigation with movable objects. Their algorithm, Look and Manipulate Backchaining (LAMB), however, relies on the assumptions of deterministic action outcomes and on extremely simplified environment dynamics (grasps always succeed, pushed objects always slide along axis without rotation), having limited applicability in the real world.

In this work, we do not focus on retrieving individual objects, but on mapping and identifying all objects within an environment. With our long-term occupancy and semantic map representation, retrieval plans can be generated without relying on perfect model knowledge or single-shot scene understanding, while our learned belief updates enable modeling of more complex manipulation behaviors.

[1] https://github.com/NilsDengler/manipulation_enhanced_map_prediction

C. Learned World Dynamics Models

Many model-based reinforcement learning algorithms learn environment models from episodic environmental interaction, often in latent spaces for improved evaluation speed [14]. These models, however, do not result in a human-interpretable representation in contrast to our learned map-space dynamics. Another recent trend leverages the popularity of conditional video diffusion models to develop interactive "pixel-space" simulators [45, 43]. These models focus on short-term visual fidelity. We posit that explicit 3D maps provide long-term temporal consistency and calibrated uncertainty measures that are needed for robotic tasks that involve information gathering.

III. Problem Definition

We address the MEM problem as follows: Consider an environment with a set of movable objects of varying sizes and orientations, where some objects may be occluded and not directly observable from any viewpoint. The arrangement of these objects, along with the fixed support geometry (e.g., a shelf or table), constitute the workspace configuration space C^w [5]. The environment's initial configuration is unobservable and denoted $c_0 \in C^w$.

The robot's objective is to create an accurate representation of the workspace configuration c_t after the execution of a sequence ζ of actions $[a_0, \ldots, a_n]$. These actions include two types: **observation actions** where the robot moves its camera to a specific viewpoint $v_t \in \mathbb{V}$ to capture an RGB-D image, and **manipulation actions**, where the robot interacts with the scene (e.g., via pushing or grasping). We address the eye-in-hand RGB-D camera setting in which the robot does not receive informative observations during manipulation and must instead move to a retracted viewpoint to receive valid depth data due to minimum depth restrictions. Moreover, we do not integrate views during movement between locations, since such images are subject to motion blur.

To formalize the problem, let the robot's internal representation of the environment, a belief over metric-semantic maps explaining object classes over the workspace, be denoted as Φ_t. We assume a closed set of N_{classes} semantic classes. Let the most-likely map according to a belief Φ_t be denoted ϕ_t. When the robot executes a manipulation action a_t, it causes a transition $c_t \mapsto c_{t+1} \in C^w$ according to the environment's dynamics $c_{t+1} = \mathrm{Dyn}(c_t, a_t)$. Additionally, whenever the robot takes any action a_t and gets an observation o_t, drawn according to the observation function $Z(o_t|a_t, c_{t+1})$, its internal representation is updated through its belief update function $\Phi_{t+1} = \mathrm{BelUpdate}(\Phi_t, a_t, o_t)$.

Finally, similar to Choudhury et al. [7], we define MEM as an optimal budgeted mapping problem. The robot is given a maximum action budget *T*, and an initial environment configuration c_0, which is *a priori* unknown. The task is to output the most informative sequence of actions ζ such that the robot's predicted map ϕ_T, at the last step of the budget, maximizes its mean Intersection over Union (mIoU) to the ground truth map ϕ_T^{GT} which represents c_T. So we have:

$$\begin{aligned} \zeta^* \quad &= \underset{\zeta=[a_0,\cdots,a_{T-1}]}{\arg\max} \ \text{mIoU}\left(\phi_T, \phi_T^{\mathrm{GT}}\right) \\ \text{s.t.} \quad &\phi_T^{\mathrm{GT}} = \text{ToMap}(c_T) \\ &c_{t+1} = \text{Dyn}(c_t, a_t) \forall t \\ &\Phi_{t+1} = \text{BelUpdate}(\Phi_t | a_t, o_t), o_t \sim Z(o_t | a_t, c_{t+1}), \forall t \end{aligned} \tag{1}$$

where $\mathrm{BelUpdate}(\cdot)$ represents the robot's belief update, $Z(\cdot)$ is the observation function, and $\mathrm{ToMap}(\cdot)$ yields the metric-semantic map that corresponds to a known configuration of the environment. In deployment, the robot cannot accurately predict ϕ_T^{GT}, as it does not have access to the initial configuration nor the dynamics of the environment. It may not even know the number of objects, their shapes, or semantic labels.

IV. Method

A. Overview

We model the MEM problem as a Partially Observable Markov Decision Process (POMDP) in metric-semantic map-space Ω. To solve this POMDP, the agent should perform a belief update about the state of the map after both manipulation and observation actions. For manipulation actions, the belief update propagates through a map transition function $T(\phi_{t+1}|\phi_t, a_t)$ that is *a priori* unknown. For observation actions, the belief update should integrate the observation while reasoning over hidden object shapes and arrangements to reduce uncertainty.

However, the high dimensionality of the space of all possible maps makes traditional belief updates computationally infeasible [20]. The approximation that map cells are independent, as commonly done in occupancy grid mapping [41], leads to compact belief representations but reduces precision in the belief update. Our CNABU method captures prior knowledge about the world, such as object contiguity and usual object sizes and arrangements, in its update. We also leverage uncertainty-aware deep learning models to predict factorized belief updates that are better aligned with plausible map configurations and yield actionable quantification of uncertainty.

These models are trained using simulated ground truth to approximate occlusion reasoning and interaction dynamics, *i.e.*, Dyn. Object sizes, classes, occlusion levels, and manipulation effects should be roughly representative of real scenes, but our method is tolerant to differences in configuration, number, class distribution, and moderate shape changes.

Finally, our POMDP solver approximates the MEM objective with the Volumetric Information Gain (VIG) metric [10], since directly modeling the mIoU metric is not feasible, as it depends on the unobservable workspace configuration. It uses a 2-step greedy approach that obtains good performance by exploiting near-submodularity of the VIG function. Moreover, the 2-step approach obviates the need to sample from the observation distribution Z.

B. Neural Map Belief Dynamics

Following grid mapping literature [41], we represent a belief Φ_t over the semantic-metric map using a Bernoulli distribution

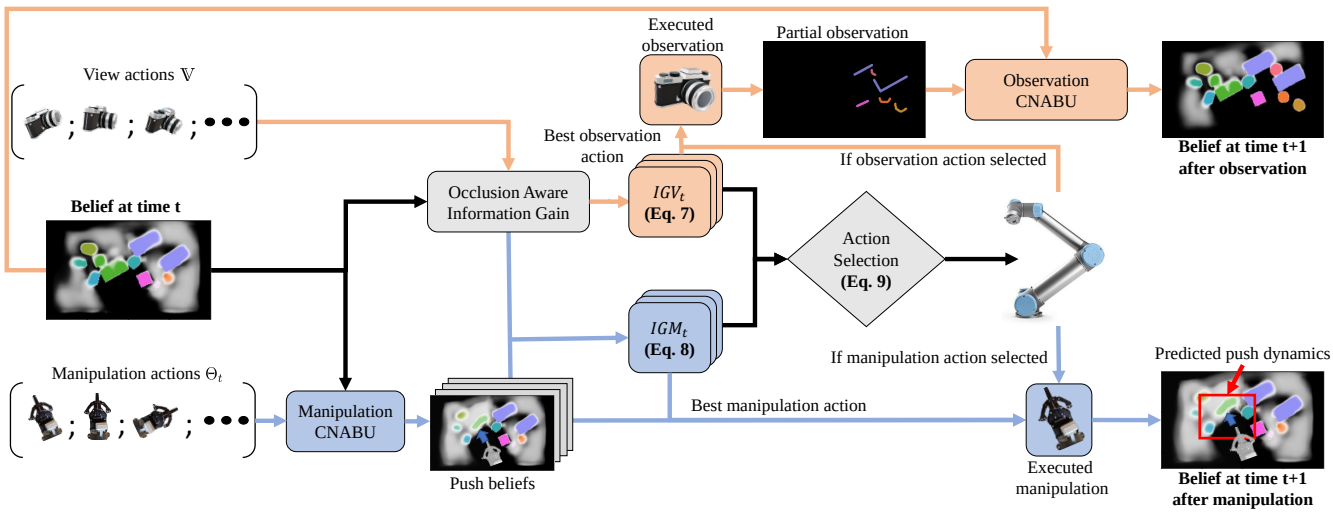


Fig. 2: From a prior map belief, our pipeline predicts a map belief resulting from a set of candidate pushes. It then weighs the information gain from taking two consecutive independent views given the current belief (orange arrows) or taking a single observation given any of the predicted beliefs after pushing (blue arrows), selecting the path of highest cumulative information gain and taking its respective first action – either taking the next best view or executing the best push. IGV_t represents the best information gain obtainable from taking two distinct observation actions, whereas IGM_t is the best information gain obtainable through a manipulation action followed by an observation action.

for a cell's occupancy and a categorical distribution for a cell's semantic class. Each cell is assumed independent. We represent the occupancy map as a 3D voxel belief $\Phi_t^O \in \mathbb{R}^{H\times W\times D}$ and the semantic map as a 2D birds-eye belief $\Phi_t^S \in \mathbb{R}^{H\times W\times N_{classes}}$. The semantic map is 2D for simplicity because objects are roughly prismatic and stacking is not allowed in our problem, while the occupancy is 3D because object heights affect visibility determination. We consider observations $o_t \in O$ consisting of an RGB-D image with added semantic labels.

To perform efficient belief updates, we introduce a Calibrated-Neural Accelerated Belief Update (**CNABU**) technique that uses separate neural networks to represent the posteriors of the viewpoint and manipulation actions in the factored representation. The first, called *observation CNABU*, computes a map belief update after a observation action $\Phi_{t+1} \leftarrow \sigma_o(\Phi_t, o_t, a_t)$. The second, called *manipulation CNABU*, computes a map belief update after a manipulation action $\Phi_{t+1} \leftarrow \sigma_m(\Phi_t, a_t)$, where we drop the observation because no new observation is generated.

Let $\Phi_t(\phi) = P(\phi|\Phi_t)$ denote the probability density of a map ϕ under belief Φ_t. For any action a_t and observation o_t, the standard POMDP belief update equation [20] gives the posterior belief as:

$$\Phi_{t+1}(\phi) = \frac{1}{\eta} Z(o_t|a_t,\phi) \sum_{\phi'\in\Omega} T(\phi|\phi',a_t)\Phi_t(\phi'), \qquad (2)$$

where ϕ' ranges over all possible maps, with η being a normalizing constant. If we wished to project the belief state into a marginalized occupancy probability of a cell, we would need to compute:

$$\Phi_{t+1}^O[i,j,k] = \mathbb{E}_{\phi\sim\Phi_{t+1}}[\phi^O[i,j,k]=1], \qquad (3)$$

where $\mathbb{E}$ is the expected value. A similar equation would hold for semantic updates. Regardless, the space of possible maps is far too large [41] and there is no current method for assessing map densities $\Phi_t(\phi)$ that accurately accounts for inter-cell correlations (e.g., object shapes, arrangements, and visibility).

Instead, we train CNABUs through simulation data. Since any scene that could produce a belief Φ_t via its observations is inherently a sample of the distribution induced by this belief, the training process leverages a neural network's averaging tendency to create an implicit Monte Carlo estimate of Eq. 3. We postpone the network and training details to Section. IV-G and proceed to describe their use in MEM.

C. Solving the POMDP

We propose to solve the map-space POMDP by using a k-step receding horizon greedy planner, as shown in Fig. 2, which uses Volumetric Information Gain (VIG) [10] as an approximation of the true reward. VIG correlates to information gain and is a submodular optimization objective in static scenes [23], having been used to efficiently solve NBV planning and sensor placement problems. Due to VIG's submodularity, a greedy policy for solving this problem leads to bounded suboptimality, justifying the greedy receding-horizon strategy. While VIG's submodularity does not hold in general for dynamic scenes, we assume that manipulation actions do not increase entropy at a rate that justifies the significant expense of long lookaheads. However, because a manipulation action does not produce any observation (and hence no immediate information gain) at least one further observation must be considered in scoring. Hence, the minimum viable search horizon at $k = 2$, which yields a balanced tradeoff between computational efficiency and action quality.

To perform an observation action, the robot chooses from $v^i \in \mathbb{V}$ possible views in a fixed array of camera positions $\mathbb{V}$ to which the robot can move. Furthermore, let $\theta_t \in \Theta_t$ be a sampled manipulation action from a set of feasible manipulation actions. In our two-step greedy search, we only

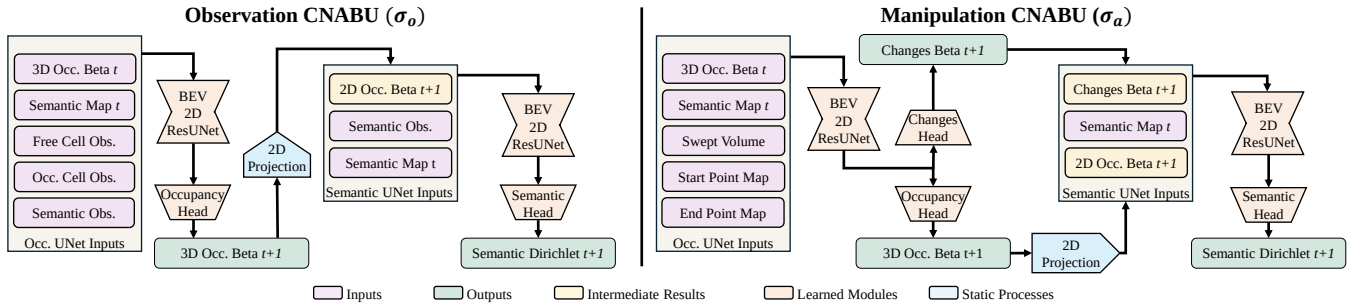


Fig. 3: Architecture overview of our observation and manipulation CNABU networks. The observation prediction network uses the occupancy posterior beta and semantic posterior Dirichlet for loss computation, while the manipulation prediction network additionally takes the 2D map of occupancy changes after the push for loss calculations.

consider two possible kinds of action sequences: taking two observation actions (v_t, v_{t+1}) or performing a manipulation action followed by an observation (θ_t, v_{t+1}). This is because (θ_t, θ_{t+1}) would result in no observation and therefore no information gain and the information gain of (v_t, θ_{t+1}) is strictly smaller than the VIG of any (v_t, v_{t+1}), $v_{t+1} \neq v_t$.

Letting $IG(v_i, \cdots, v_{i+n-1}|\Phi^O)$ denote the Volumetric Occlusion-aware Information Gain [10] for voxels intersected by the rays from n views $(v_i \cdots v_{i+n-1})$ given belief Φ^O , the two most informative consecutive views (v_t^*, v_{t+1}^*), are:

$$(v_t^*, v_{t+1}^*) = \underset{v_t, v_{t+1} \in \mathbb{V}}{\arg\max}\, IG(v_t, v_{t+1}|\Phi_t^O) \tag{4}$$

For manipulation, $\tilde{\Phi}_{t+1}^{\theta_t} = \sigma_m(\Phi_t, \theta_t)$ denotes the predicted belief from the manipulation CNABU when given action $\theta_t \in \Theta_t$ as input. We can define the most informative 1-step push, θ_t^* and its associated most informative view $v_{\theta_t}^*$, as:

$$\left(\theta_t^*, v_{\theta_t}^*\right) = \underset{\theta_t \in \Theta_t, v_{t+1} \in V}{\arg\max}\, IG(v_{t+1}|\tilde{\Phi}_{t+1}^{\theta_t}) \tag{5}$$

Finally, let $H(\cdot)$ represent the entropy of a given semantic map, with the entropy change defined as $\Delta H(\Phi_t, \Phi_{t+1}) = H(\Phi_t) - H(\Phi_{t+1})$. The term $IGV_t = IG(v_t^*, v_{t+1}^*|\Phi_t^O)$ denotes the best information gain obtained from two viewpoints, also called **ViewPoint Planning** (VPP), while $IGM_t = IG(v_{\theta_t}^*|\tilde{\Phi}_{t+1}^{\theta_t^*})$ represents the best information gain from a push followed by a viewpoint, which we will call **push selection**. Lastly, $Reg_t = \Delta H(\Phi_t, \tilde{\Phi}_{t+1}^{\theta_t^*})$ captures the entropy difference between the current semantic map and the map after the best push.

Our policy decides the action a_t to take according to:

$$a_t = \begin{cases} v_t^* & \text{if } IGV_t > IGM_t + \gamma Reg_t \\ \theta_t^* & \text{otherwise} \end{cases} \tag{6}$$

Where γ is a discount factor to account for different magnitudes between camera array VIGs and whole-map semantic entropy values. Reg_t serves as a regularization on the aggressiveness of the selected pushes, as more radical environment manipulations could potentially reveal more of the environment, but would cause unnecessary disturbances to the scene and introduce a lot more uncertainty to the post-push representations. If $a_t = \theta_t^*$, no observation is taken and $\Phi_{t+1} = \tilde{\Phi}_{\theta_t^*, t+1}$. If a_t is an observation action, we get the observation o_t and use σ_o to obtain the new belief $\Phi_{t+1} = \sigma_o(\Phi_t, a_t, o_t)$. This search is then repeated in a loop until the maximum number of actions has been performed, or a threshold for full map completion has been reached, which we set to 95% of the semantic voxels with greater than 85% certainty in the belief Φ. When this threshold is reached, the planner no longer pushes, but still collects novel views.

D. Push Sampling

We consider pushing as our manipulation action. To compute valid push candidates using Φ_t^O, we first compute the high-confidence frontier points from the shelf entrance 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 other high confidence voxels in Φ_t^O. After randomly sampling these k unique frontier points, we determine for each point whether this starting position of a push will lead to a feasible and valid motion plan. For each valid start point, we sample a likely occupied point in Φ_t^O 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 using a sampling based motion planner [13, 15] and parametrize this plan with θ.

E. Evidential Posterior Learning

Although we could learn to predict beliefs as functions of Φ, we introduce an enhanced representation that uses evidential posterior networks [42], as evidential learning is known improve the calibration of uncertain predictions. An *evidential map belief* $\boldsymbol{\lambda}$ consists of belief prior parameters for each cell of the map belief Φ. Specifically, we store $\boldsymbol{\lambda^O} \in \mathbb{R}^{H \times W \times D \times 2}$, a 3D grid of Beta distribution parameters for each voxel in the map, and $\boldsymbol{\lambda^S} \in \mathbb{R}^{H \times W \times N_{\text{classes}}}$, a grid of Dirichlet distribution parameters for each cell in the 2D map. Let $\text{Beta}(\cdot)$ and $\text{Dir}(\cdot)$ denote the Dirichlet and Beta distributions, respectively. Therefore, the occupancy and semantic map beliefs are related to evidential parameters via $\Phi^O[\cdot] = \mathbb{E}[\text{Beta}(\boldsymbol{\lambda^O}[\cdot])]$ and $\Phi^S[\cdot] = \mathbb{E}[\text{Dir}(\boldsymbol{\lambda^S}[\cdot])]$. We assume that the initial states $\boldsymbol{\lambda_0^O}$ and $\boldsymbol{\lambda_0^S}$ are uninformed and

set to $\mathbb{1}$, a unit tensor with appropriate dimensions. Note: evidential parameters are maintained for each map belief Φ in our algorithm and CNABUs operate by propagating the evidential parameters ($\boldsymbol{\lambda_{t+1}} \leftarrow \sigma_o(\boldsymbol{\lambda_t}, o_t, a_t)$ and $\boldsymbol{\lambda}_{t+1} \leftarrow \sigma_m(\boldsymbol{\lambda_t}, a_t)$) followed by an update to the standard belief parameters.

F. Dataset Generation

To train the CNABU models σ_o and σ_m, we collect datasets on maps, viewpoints, and sampled pushes in simulation. A total of 14 different object categories from the YCB dataset [4] 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 objects. This method allows for the sampling of varied object configurations, numbers and classes, and is described in more detail in Appendix A.

To train the viewpoint belief prediction model σ_o, we assemble a dataset $\mathbb{D} = \{d_1, d_2, \ldots\}$ where each scene d_i has the form $(\phi^{\text{GT}}, o_1, \cdots, o_n)$. Here, ϕ^{GT} represents the ground truth 3D metric-semantic voxel map of a shelf environment with randomly placed objects, and $o_1, \cdots, o_n$ are the depth and semantically segmented images captured from $\mathbb{V}$, the set of discrete viewpoints in the environment. The ground truth semantic labels are used in the rendered images.

To train the manipulation belief prediction model σ_m, the simulated robot executes a randomly sampled action in synthesized scenes. This produces a dataset $\mathbb{D}^a = \{d_1^a, d_2^a, \ldots\}$ with each sequence $d_i^a = (\phi_{\text{pre}}^{\text{GT}}, \phi_{\text{post}}^{\text{GT}}, o_1, o_2, \cdots, o_n, a)$, where $\phi_{\text{pre}}^{\text{GT}}, \phi_{\text{post}}^{\text{GT}}$ are the ground truth maps before and after manipulation, respectively, a is the executed action and $o_1, \cdots, o_n$ are observations from $\mathbb{V}$ as before, taken before the manipulation is executed.

G. Training CNABU Networks

We now outline the procedure for training the two CNABUs, whose network architectures are shown in Fig. 3, with further details given in Appendix B.

We train σ_o from the dataset $\mathbb{D}$ by sampling, without replacement, a sequence of l posed depth and semantic images, $d' = (o'_0, \cdots, o'_{l-1})$, from every scene d_i at every epoch. This sampling diversifies the beliefs encountered during training by varying the emulated observation sequences for each scene.

The loss over the sequence d' is computed as follows. We recursively predict the evidential beliefs $\boldsymbol{\lambda_t} = \sigma_o(\boldsymbol{\lambda_{t-1}}, o'_{t-1})$ up to time l. Also, let $y^i \in \{0,1\}^{\dim}$ indicate a one-hot encoded tensor of the ground truth value for a given voxel i according to ϕ^{GT}, where dim is either N_{classes} for $\lambda_{\mathbf{t}}^{\mathbf{S}}$ and 2 for $\lambda_{\mathbf{t}}^{\mathbf{O}}$. For each voxel i in ϕ_t, define, for its predicted distribution parameter $\lambda_t^i \in \boldsymbol{\lambda}_t$, $\tilde{\lambda}_t^i = y^i + (1 - y^i) \odot \lambda_t^i$, where $\odot$ denotes element-wise multiplication. We employ the evidential uncertainty-aware cross-entropy from Sensoy et al. [38] as the loss, which, for each time step, is given by:

$$L_t^{\text{type}}(\lambda_{\mathbf{t}}^{\text{type}}, \phi^{\text{GT}}) = \sum_{\lambda_t^i \in \lambda_{\mathbf{t}}^{\text{type}}} \mathcal{L}(\lambda_t^i, y^i) + \varepsilon \text{KL}\left(Dir(\tilde{\lambda}_t^i) || Dir(\mathbb{1})\right) \quad (7)$$

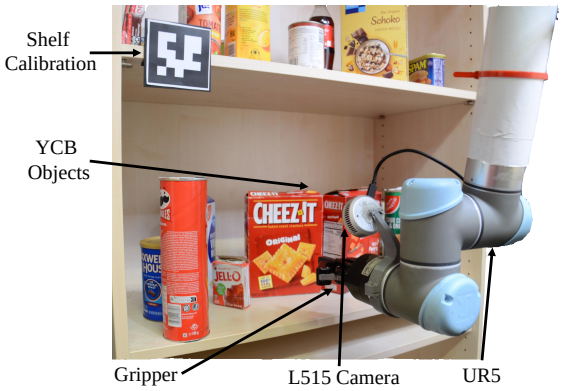


Fig. 4: Real-world environment showing a shelf scenario. The UR5 is equipped with an Robotiq parallel-jaw gripper and a Realsense L515 RGB-D camera to create a calibrated representation of the scene.

, where $\mathcal{L}(\lambda_t^i, y^i) = \sum_{j=0}^{\dim} y^{ij} \left(\log(S_t^i) - \log(\lambda_t^{ij})\right)$, $S_t^i = \sum_{j=0}^{\dim} \lambda_t^{ij}$, $\text{KL}(a||b)$ is the Kullback-Liebler divergence between distributions a and b, $\mathbb{1}$ is a vector of all ones, $type$ is o if it is an occupancy loss (dim = 2) and s if it is a semantic loss (dim = N_{classes}), and ε is an annealing parameter, set according to Sensoy et al. [38]. The total loss for the sample d' is the sum of the semantic and occupancy losses $L_t^o + L_t^s$ over the l observations.

The manipulation CNABU is defined similarly to the observation CNABU, except that it has an auxiliary output, which predicts a Beta distribution over a voxel grid modeling the probability of a given voxel being changed in Φ^{GT} after the manipulation is executed, which we call $\boldsymbol{\lambda}^{\mathbf{diff}}$. Therefore, we have $\boldsymbol{\lambda}_{\boldsymbol{t+1}}^{\boldsymbol{S}} \boldsymbol{\lambda}_{\boldsymbol{t+1}}^{\boldsymbol{O}}, \boldsymbol{\lambda}_{\boldsymbol{t+1}}^{\mathbf{diff}} = \sigma_m(\boldsymbol{\lambda}_t, a_t)$

Training epochs iterate over sequences in $\mathbb{D}^a$. For each sequence, we sample a subsequence of $l \in [1, 10]$ images without replacement as above. We then recursively obtain the beliefs from the observation CNABU, $\boldsymbol{\lambda}_{t+1} = \sigma_o(\boldsymbol{\lambda}_t, o'_t)$ for $t = 0, \ldots, l-1$. Next, we predict the post-manipulation belief $\boldsymbol{\lambda}_{\boldsymbol{t+1}}^{\boldsymbol{S}} \boldsymbol{\lambda}_{\boldsymbol{t+1}}^{\boldsymbol{O}}, \boldsymbol{\lambda}_{\boldsymbol{t+1}}^{\mathbf{diff}} = \sigma_m(\boldsymbol{\lambda}_l, a)$ and use $\phi_{\text{post}}^{\text{GT}}$ as the training target. The random sampling of a different number of observations prior to pushing ensures the CNABU sees different belief stages during training. We derive the ground truth for $\boldsymbol{\lambda}^{\mathbf{diff}}$, $\phi_{\text{change}}^{\text{GT}}$, from the difference between $\phi_{\text{pre}}^{\text{GT}}$ and $\phi_{\text{post}}^{\text{GT}}$.

Finally, we add a fourth loss component, which we call consistency loss, L^{con} which is the Mean Squared Error between $\boldsymbol{\lambda_{l+1}}$ and $\boldsymbol{\lambda_l}$. This loss serves as a regularization to encourage the alpha parameters of the distributions in the unchanged areas of the map to have a similar magnitude. As before, the network heads are trained using the loss in Eq. 7 [38]. The total loss for the manipulation sequence is given by $L = L_{l+1}^O + L_{l+1}^S + L_{l+1}^{\text{diff}} + \epsilon L^{\text{con}}$.

V. Experiments

We perform four core experiments to evaluate our approach. First, we test in simulation to highlight our pipeline's improvements in map completeness and accuracy compared to state-of-the-art [11]. Next, we present a series of ablations of our method and evaluate several interactive baselines. We

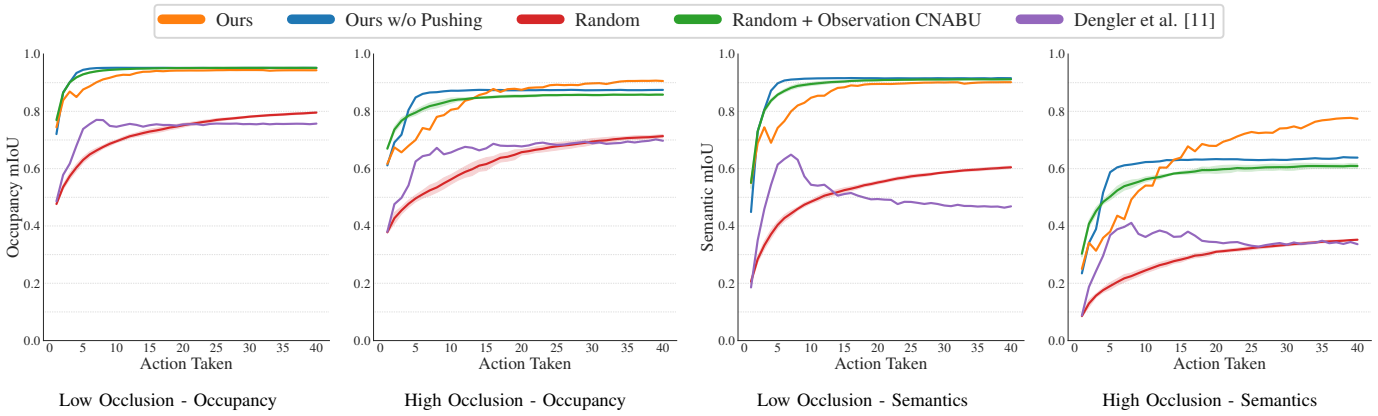


Fig. 5: Simulation results for Manipulation Enhanced Mapping against SOTA and non-pushing baselines—showing both occupancy and semantic IoUs over time for each method. Our method outperforms all baselines in highly occluded scenes, while not having degraded performance in low occlusion scenes. Standard deviation of performance of random baselines over random seeds is represented as shading around each plot.

then present some experiments on the generalization of the trained CNABUs to within-class shape variations. Finally, we study the robustness of our system in terms of its zero-shot transferability to a physical setup. Further evaluations, which validate the individual CNABU's performance and the use of VIG as a reward proxy, are provided in Appendices C and D.

A. Experimental Setup

Our task setup consists of a shelf scene with a UR5 arm for observation and action execution (Fig. 4). The robot is fixed to a table facing an occluded shelf and equipped with a Robotiq parallel-jaw gripper for manipulation and an RGB-D camera for observations. In simulation, the ground truth observations and segmentations are provided by rendering.

The real-world setup is similar, but with a few notable differences. For action execution, ROS and MoveIt [6] are used. The depth image is obtained from an intel Realsense L515 camera and the semantic segmentation in the real world is performed using segment anything 2 (SAM2) [35] and a strategy similar to LSeg [25, 19]. We take detected masks from SAM2 and crop the original image around them. Next, we extract their CLIP [34] embeddings and compute their cosine similarity to the language embeddings of our target classes, whose prompts we list in Appendix Sec. E. Finally, we normalize the similarity scores to classify each mask.

B. Simulation Experiments

These experiments consider both low and high occlusion scenarios for manipulation-enhanced mapping. We generate 100 low occlusion scenarios via rejection sampling, using our sampling method described in Appendix A, but keeping only scenarios for which at least one object cannot be seen from any of the 300 viewpoints. We then crafted 25 high occlusion scenarios by hand to be challengingly crowded and with many objects occluded. We provide examples of each category in Figs. 10 and 11 in the Appendix. The robot begins with a naive uniform map prior.

We compare our work (**Ours**) with the following baselines. First, we reimplemented the approach of Dengler et al. [11] and fine-tuned the network weights provided by the authors for only 5,000 action steps. Although their experimental setup closely aligns with ours, our tests introduce a lower upper shelf board height and more densely sampled object configurations. Second, the **Random** baseline randomly samples a set of unique views $[v_0^r, \ldots, v_n^r] \in \mathbb{V}$ and uses standard metric-semantic occupancy mapping [41]. We also combine random view selection with our observation belief predictor σ_o, **Random + Observation CNABU**. We also compare an ablation of our pipeline that does not use manipulation, **Ours w/o pushing**.

Metric and semantic mIoU compared to the ground truth map at time t are plotted in Fig. 5. We observe that the previous S.O.T.A. for unstructured MEM, Dengler et al. [11], explores efficiently at early stages. However, with more of the scene uncovered and new areas being harder to observe, its performance degrades to comparable or worse than random, particularly in terms of semantic mIoU. We attribute this performance degradation to two key challenges: the use of heuristic-based action switching and the lack of map updates after manipulation. The heuristic switching mechanism relies on hand-crafted rules to alternate between observation and manipulation actions, and may not always select the most informative action, occasionally choosing to push too early, too often, or to continue observing when manipulation would be more beneficial. Our POMDP-based action selection overcomes this problem and does not select manipulation actions when they are not beneficial for higher information gain. Furthermore, because the Dengler et al. [11]'s pipeline does not update its belief after a push, it requires multiple subsequent observations to reconcile inconsistencies between the actual scene and the previously assumed map representation. This delay in belief correction leads to inefficient re-exploration and degraded semantic accuracy, as the agent lacks a reliable signal for where to focus its attention.

Moreover, we observe that belief prediction is a powerful approach, leading to excellent scene coverage in low occlusion scenes even without pushing. In highly occluded scenarios, pushing is required to make progress after the visible surfaces

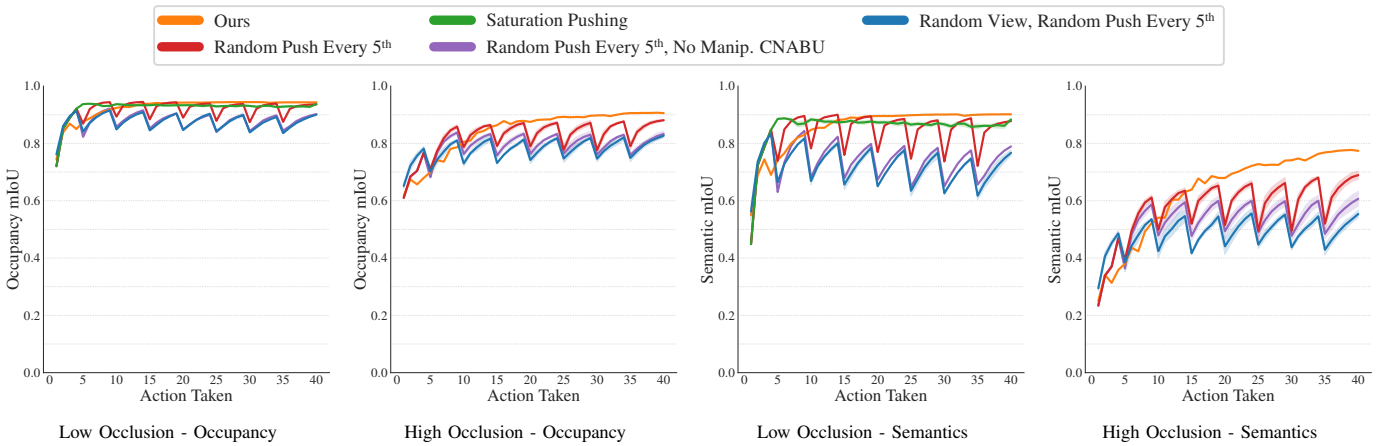


Fig. 6: Simulation tests of push selection alternatives. Note how our method not only achieves better mIoUs than any of the other methods, it does so consistently across all the steps, avoiding uninformative or overly aggressive manipulation actions. Also, next-best-view with the observation belief prediction but without push belief prediction (purple) leads to degraded performance. Standard deviation of performance of random baselines is represented as shading around each plot.

are observed. Our method uses pushing to achieve significantly higher mIoUs. Note that its IoU growth is slower early on, because pushing does not provide information until a viewpoint step is taken in the following action.

C. Push Selection Alternatives

Next, we validate our POMDP push selection strategy. Unless otherwise noted, the same belief prediction networks are provided to each method. Three of the strategies push regularly at a five-step interval, which is a typical rate at which observation actions provide diminishing returns. The first, **Random Push Every** 5^{th}, plans views using NBV and σ_o, and randomly samples a push every five steps and updates its belief using σ_m. The second, **Random Push Every** 5^{th}**, No Manip. CNABU** is the same, but does not use σ_m to update its belief after pushing. The third, **Random View Random Push Every** 5^{th}, chooses random views and randomly pushes every five steps without using σ_m. Finally, we consider a heuristic that performs a random push when VIG seems to saturate, **Saturation Pushing**. The saturation threshold is when two consecutive estimates of VIG during NBV differ by less than 2%. Saturation pushing uses σ_m after each push to update its belief. A comparison of selection strategies is given in Sec. F in the Appendix.

Results are shown in Fig. 6. Each baseline that has a random component is run three times with different random seeds. We observe that push belief prediction is beneficial to manipulation-enhanced mapping, even when random pushes are being executed. The blue and purple curves show methods that are not informed by σ_m. Further, we see that the saturation pushing (green) does not observe post-push performance drops, but it is ultimately outperformed by random informed pushes and our method. Overall, our method still achieves the best performance, most strikingly in highly occluded scenes.

D. Out of Distribution Shape Experiments

In order to evaluate the robustness of the proposed pipeline to shape variations within a given class, we modify the simulation setup. For each individual instance of an object in a given scene, we randomly and independently rescale the object's mesh on each of its principal axis by a factor chosen uniformly at random within $[0.8, 1.2]$, and re-run our simulation experiments in the high-occlusion scenes with these out-of-distribution shapes for the strongest baselines in simulation. As seen in Fig. 8, we observe a decrease in the agent's performance when out of distribution, but note that in the long run the performance remains consistent with our ID experiments. This is notable since we do not perform any sort of shape augmentation during the training of the CNABUs, suggesting that the fact that they are grounded in observed maps helps them overcome some level of distribution shift. Naturally, we presume that adding shape augmentation and more diverse object geometries of the same class during training should further help reduce this OOD performance gap.

E. Hardware Experiments

We also performed 10 real-world experimental runs on a UR5 as described in Sec. V-A. All results are collected in a zero-shot fashion, i.e., no fine-tuning on real data was performed. We set the budget to 20 steps and sampled a fixed set of 75 reachable camera poses in front of the shelf for $\mathbb{V}$.

We handcrafted 10 challenging scenes, each with an average of 18 objects from the YCB dataset [4], where pushing is required to reveal other objects. We collect the ground truths by removing the top of the shelf at the end of each episode to manually score the final maps. We score each scenario according to the status of **all of the objects** present in the map. Each object in the map is classified in four categories: 1) **Correctly Found** if the majority of the object is correctly represented in the map with the right class; 2) **Misclassified But Found** if the majority of the object is present in the occupancy map but is mislabeled; 3) **Not Found** - if the majority of the object is absent from the occupancy map and 4) **Hallucinated** if an object that is not present in the scene is present in the map. Due to the complexity of precisely resetting a scene multiple times for different baselines and the fact that

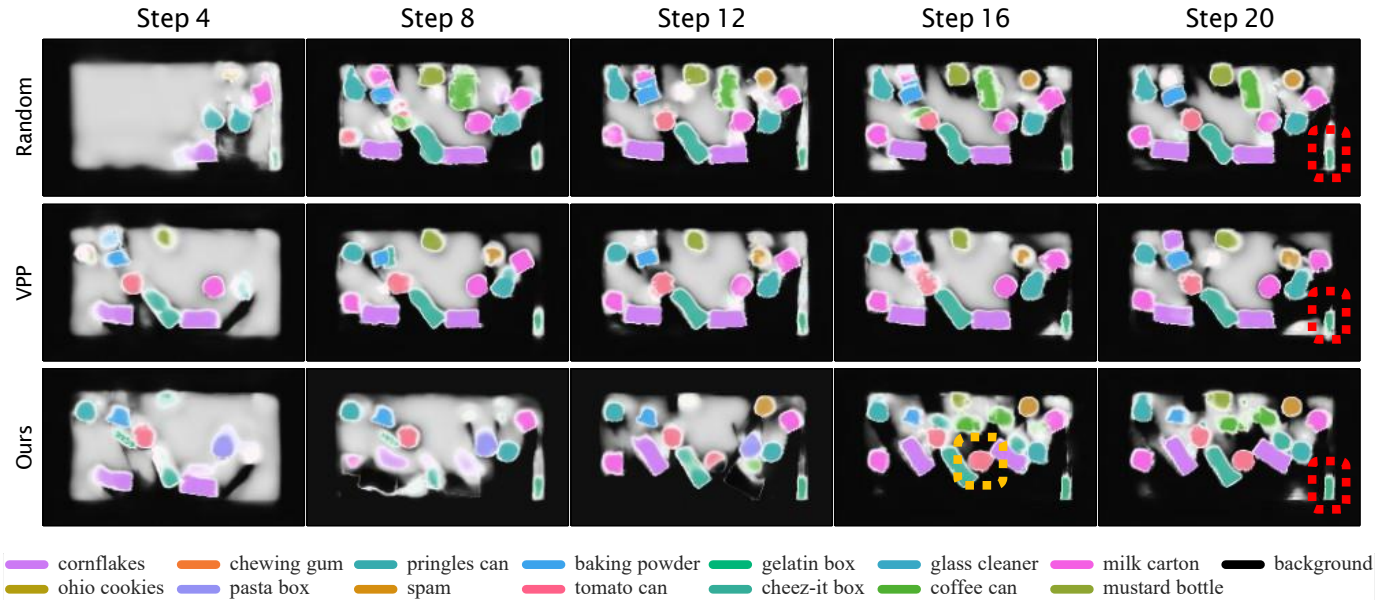
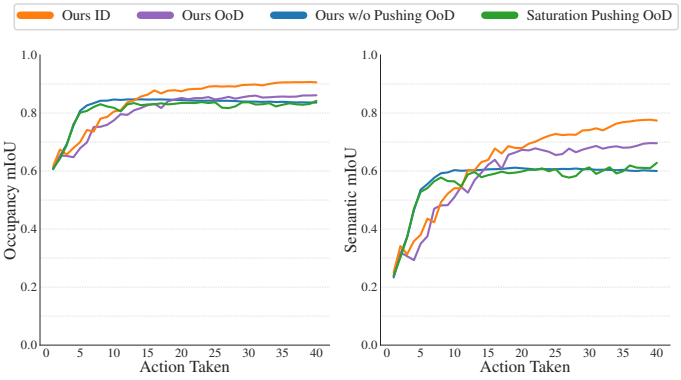


Fig. 7: Qualitative real-world experiment results. Note how VPP and Random baselines are unable to fully explore the environment due to its occlusions, while our method is able to better explore it via reasonable manipulations. In Step 16, Ours, we highlight in yellow one of the scene objects revealed by manipulation. We highlight in red a persistent hallucination across all 3 methods, likely due to unreliable semantic segmentation and significant camera noise at that corner of the shelf .



(a) Occupancy Results (IoU)

(b) Semantic Results (mIoU)

Fig. 8: OoD Results in *High Occlusion* scenes.

both random + observation CNABU and our approach without pushing are about as strong as other pushing baselines (see Figs. 5 and 6), we only compare against them in the physical experiments. For each model, we report the total quantity of each detection at time step 20 summed over all 10 trials.

Results in Tab. I show that with zero-shot transfer from sim-to-real, the proposed method still manages to retain its edge over the compared baselines. Note that all methods compared use calibrated belief prediction. However, little difference is seen between viewpoint planning (Ours w/o Pushing) and random viewpoint choices. We expect that this is due to a domain gap caused by camera noise from the realsense L515 leading to some strong artifacting in the depth images and the inaccuracies of the open-set semantic segmentation pipeline. We found the SAM2 segmenter performed particularly poorly for oblique angles or partial object views, resulting in many missed instance detections and misclassifications. For instance, there is a consistent artifact on the segmentation pipeline, which consistently classifies the bottom right corner of the shelf as a gelatin box, as seen in all methods in Fig. 7. Nevertheless, we can see that our methods (both with pushing and without pushing) greatly reduce the number of hallucinations and improve the number of correctly identified objects. Further, our complete pipeline reveals 39% of the objects that were previously unseen by the non-interactive baselines, performance consistent with the simulation experiments, despite the significant sim-to-real gap, particularly in segmentation performance, leading to many of the newfound objects being incorrectly classified.

TABLE I: Comparing our method to the strongest baselines in zero-shot transfer to real-world shelves. Total counts of each type of detection are reported across our 10 trials.

Policy	Correctly Found ↑	Missclassified But Found↑	Not Found↓	Hallucinated↓
Random	72	47	52	11
Ours w/o Pushing	81	38	52	**6**
Ours	**85**	**52**	**35**	7

VI. Limitations

Limitations of our method include the need for representative simulation training data or ground truth segmented maps. It also relies on high-quality semantic segmentation, and although the computer vision field is making significant progress on segmentation, segmentation accuracy is still too low for many robotics applications in occluded, poorly lit and partial views, especially in open-set scenarios.

Computation times for our POMDP solver vary but take on the order of several seconds, due to the need for information gain calculations and belief propagation for many

actions. Our current framework naïvely samples manipulation actions during action selection, and more intelligent action sampling could improve computational efficiency. This will be especially important when including additional manipulation actions, e.g., grasping.

There is currently a significant sim-to-real gap that could be addressed by fine-tuning on real data or performing domain randomization of the object dynamics during data collection in simulation to help improve real-world performance. Further, our maps are defined over dense voxel grids, which poses scalability challenges when applied to larger spaces. Moreover, the mapped region is a fixed volume and we assume prior knowledge about the fixed parts of the environment (e.g., shelf) for motion planning. These assumptions should be relaxed to address fully unstructured and unknown worlds. Finally, our maps use a closed-world semantic labeling, and extending belief propagation to open-world segmentation would be an interesting frontier to explore.

VII. Conclusion

This paper presented a POMDP approach to the manipulation-enhanced mapping problem in which the solver decides between changing a camera view and manipulating objects to map an area cluttered with objects. It relies on the novel Calibrated Neural-Accelerated Belief Update map-space belief propagation approach, which allows a unified treatment of both viewpoint change and manipulation actions. Using well-calibrated beliefs allows the POMDP solver to make decisions between different action types according to the most informative outcome. Experimental results in both simulation and on a real robot show that our approach outperforms non-manipulating and heuristic manipulation baselines in terms of occupancy and semantic map accuracy. Overall, this work offers a promising new framework for robots navigating and manipulating real-world cluttered environments.

VIII. Acknowledgments

This work used the NCSA Delta GPU cluster through allocation CIS240410 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program [2], which is supported by U.S. National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296. This work has partly been supported by the European Commission under grant agreement numbers 964854 (RePAIR) and by the BMBF within the Robotics Institute Germany, grant #16ME0999. This work was also partially supported by NIFA/USDA Awards #2020-67021-32799 and #2021-67021-34418.

References

[1] Andreas Bircher, Mina Kamel, Kostas Alexis, Michael Burri, Philipp Oettershagen, Sammy Omari, Thomas Mantel, and Roland Siegwart. Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots. *Autonomous Robots*, 40 (6):1059–1078, 2016.

[2] Timothy J. Boerner, Stephen Deems, Thomas R. Furlani, Shelley L. Knuth, and John Towns. ACCESS: Advancing innovation: NSF's advanced cyberinfrastructure coordination ecosystem: Services & support. In *Practice and Experience in Advanced Research Computing 2023: Computing for the Common Good*, PEARC '23, page 173–176, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450399852. doi: 10.1145/3569951.3597559. URL https://doi.org/10.1145/3569951.3597559.

[3] Jeannette Bohg, Karol Hausman, Bharath Sankaran, Oliver Brock, Danica Kragic, Stefan Schaal, and Gaurav S Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 2017. doi: 10.1109/TRO.2017.2721939. URL https://ieeexplore.ieee.org/document/8007233.

[4] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Benchmarking in manipulation research: The YCB object and model set and benchmarking protocols. *IEEE Robotics and Automation Magazine*, 2015.

[5] J Chase Kew, Brian Ichter, Maryam Bandari, Tsang-Wei Edward Lee, and Aleksandra Faust. Neural Collision Clearance Estimator for Batched Motion Planning. In Steven M LaValle, Ming Lin, Timo Ojala, Dylan Shell, and Jingjin Yu, editors, *Algorithmic Foundations of Robotics XIV*, pages 73–89, Cham, 2021. Springer International Publishing. ISBN 978-3-030-66723-8.

[6] Sachin Chitta, Ioan Sucan, and Steve Cousins. MoveIt! [ROS Topics]. *IEEE Robotics & Automation Magazine*, 19(1):18–19, 3 2012. ISSN 1070-9932. doi: 10.1109/MRA.2011.2181749. URL http://ieeexplore.ieee.org/document/6174325/.

[7] Sanjiban Choudhury, Ashish Kapoor, Gireeja Ranade, and Debadeepta Dey. Learning to gather information via imitation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 908–915, 2017. doi: 10.1109/ICRA.2017.7989112.

[8] Erwin Coumans and Yunfei Bai. PyBullet, a Python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2021.

[9] Michael Danielczuk, Andrey Kurenkov, Ashwin Balakrishna, Matthew Matl, David Wang, Roberto Martín-Martín, Animesh Garg, Silvio Savarese, and Ken Goldberg. Mechanical search: Multi-step retrieval of a target object occluded by clutter. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019. doi: 10.1109/ICRA.2019.8794143. URL https://ieeexplore.ieee.org/document/8794143.

[10] Jeffrey Delmerico, Stefan Isler, Reza Sabzevari, and Davide Scaramuzza. A comparison of volumetric information gain metrics for active 3D object reconstruction. *Autonomous Robots*, 42(2):197–208, 2018. ISSN 1573-7527. doi: 10.1007/s10514-017-9634-0. URL https://doi.org/10.1007/s10514-017-9634-0.

[11] Nils Dengler, Sicong Pan, Vamsi Kalagaturu, Rohit Menon, Murad Dawood, and Maren Bennewitz. Viewpoint push planning for mapping of unknown confined spaces. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1178–1184. IEEE, 2023.

[12] Georgios Georgakis, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, and Kostas Daniilidis. Learning to Map for Active Semantic Goal Navigation. In *International Conference on Learning Representations*, 2022. doi: 10.48550/arXiv.2106.15648. URL https://openreview.net/forum?id=swrMQttr6wN.

[13] Sanchez Gildardo and Jean-Claude Latombe. A single-query bi-directional probabilistic roadmap planner with lazy collision checking. In Jarvis Raymond Austin and Alexander Zelinsky, editors, *Robotics Research*, pages 403–417, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-36460-3.

[14] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2020.

[15] K. Hauser. Robust contact generation for robot simulation with unstructured meshes. In *Proc. of the Intl. Symposium on Robotic Research (ISRR)*, 2013.

[16] Benjamin Hepp, Debadeepta Dey, Sudipta N Sinha, Ashish Kapoor, Neel Joshi, and Otmar Hilliges. Learn-to-score: Efficient 3d scene exploration by predicting view utility. In *Proceedings of the European conference on computer vision (ECCV)*, pages 437–452, 2018. URL https://link.springer.com/chapter/10.1007/978-3-030-01267-0_27.

[17] H. Hu, S. Pan, L. Jin, M. Popović, and M. Bennewitz. Active implicit reconstruction using one-shot view planning. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024. doi: 10.1109/ICRA57147.2024.10611542. URL https://ieeexplore.ieee.org/document/10611542.

[18] Huang Huang, Marcus Dominguez-Kuhne, Vishal Satish, Michael Danielczuk, Kate Sanders, Jeffrey Ichnowski, Andrew Lee, Anelia Angelova, Vincent Vanhoucke, and Ken Goldberg. Mechanical search on shelves using lateral access x-ray. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2045–2052. IEEE, 2021. doi: 10.1109/IROS51168.2021.9636629. URL https://ieeexplore.ieee.org/document/9636629.

[19] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, Joshua B Tenenbaum, Celso Miguel de Melo, Madhava Krishna, Liam Paull, Florian Shkurti, and Antonio Torralba. ConceptFusion: Open-set multimodal 3D mapping, 2023.

[20] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, 1998. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(98)00023-X. URL https://www.sciencedirect.com/science/article/pii/S000437029800023X.

[21] Seungyeon Kim, Young Hun Kim, Yonghyeon Lee, and Frank C Park. Leveraging 3d reconstruction for mechanical search on cluttered shelves. In *7th Annual Conference on Robot Learning*, 2023. URL https://proceedings.mlr.press/v229/kim23a/kim23a.pdf.

[22] Michael Krainin, Brian Curless, and Dieter Fox. Autonomous generation of complete 3d object models using next best view manipulation planning. In *2011 IEEE international conference on robotics and automation*, pages 5031–5037. IEEE, 2011.

[23] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(2), 2008.

[24] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast Encoders for Object Detection From Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2019.

[25] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022.

[26] Jue Kun Li, David Hsu, and Wee Sun Lee. Act to see and see to act: Pomdp planning for objects search in clutter. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2016. doi: 10.1109/IROS.2016.7759839. URL https://ieeexplore.ieee.org/document/7759839.

[27] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. BEVFusion: A simple and robust lidar-camera fusion framework. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 10421–10434. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/43d2b7fbee8431f7cef0d0afed51c691-Paper-Conference.pdf.

[28] Joao Marcos Correia Marques, Albert J Zhai, Shenlong Wang, and Kris Hauser. On the overconfidence problem in semantic 3d mapping. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*. IEEE, 2024.

[29] Riccardo Monica and Jacopo Aleotti. Contour-based next-best view planning from point cloud segmentation of unknown objects. *Autonomous Robots*, 42(2):443–458, 2018. URL https://link.springer.com/article/10.1007/s10514-017-9618-0.

[30] Jose Muguira-Iturralde, Aidan Curtis, Yilun Du, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Visibility-aware navigation among movable obstacles. In

2023 IEEE International Conference on Robotics and Automation (ICRA), pages 10083–10089, 2023. doi: 10.1109/ICRA48891.2023.10160865.

[31] Joni Pajarinen, Jens Lundell, and Ville Kyrki. Pomdp planning under object composition uncertainty: Application to robotic manipulation. *IEEE Transactions on Robotics*, 39(1):41–56, 2023. doi: 10.1109/TRO.2022.3188168.

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H Wallach, H Larochelle, A Beygelzimer, F d´Alché Buc, E Fox, and R Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[33] Thomas Pitcher, Julian Förster, and Jen Jen Chung. Reinforcement learning for active search and grasp in clutter. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2024. doi: 10.1109/IROS58592.2024.10801366. URL https://ieeexplore.ieee.org/document/10801366.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[35] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[36] Dhruv Saxena and Maxim Likhachev. Planning for manipulation among movable objects: Deciding which objects go where, in what order, and how. In *Proc. of the Int. Conf. on Automated Planning and Scheduling (ICAPS)*, 2023.

[37] Dhruv Mauria Saxena and Maxim Likhachev. Improved M4M: Faster and richer planning for manipulation among movable objects in cluttered 3d workspaces. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611234. URL https://ieeexplore.ieee.org/document/10611234.

[38] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, and R Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/a981f2b708044d6fb4a71a1463242520-Paper.pdf.

[39] Satvik Sharma, Kaushik Shivakumar, Huang Huang, Lawrence Yunliang Chen, Ryan Hoque, Brian Ichter, and Ken Goldberg. Open-world semantic mechanical search with large vision and language models. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=vsEWu6mMUhB.

[40] Mike Stilman, Jan-Ullrich Schamburek, James Kuffner, and Tamim Asfour. Manipulation planning among movable obstacles. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*. IEEE, 2007.

[41] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Probabilistic robotics. 2005. *Massachusetts Institute of Technology, USA*, 2005.

[42] Dennis Ulmer, Christian Hardmeier, and Jes Frellsen. Prior and posterior networks: A survey on evidential deep learning methods for uncertainty estimation. *Proceedings of Machine Learning Research*, 2023.

[43] Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines, 2024. URL https://arxiv.org/abs/2408.14837.

[44] Yuchen Xiao, Sammie Katt, Andreas ten Pas, Shengjian Chen, and Christopher Amato. Online planning for target object search in clutter under partial observability. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*. IEEE, 2019. doi: 10.1109/ICRA.2019.8793494. URL https://ieeexplore.ieee.org/document/8793494.

[45] Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=sFyTZEqmUY.

[46] Rui Zeng, Yuhui Wen, Wang Zhao, and Yong-Jin Liu. View planning in robot active vision: A survey of systems, algorithms, and applications. *Computational Visual Media*, 2020. URL https://link.springer.com/article/10.1007/s41095-020-0179-3.

[47] Albert J Zhai and Shenlong Wang. PEANUT: Predicting and navigating to unseen targets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10926–10935, October 2023. doi: 10.1109/ICCV51070.2023.01003. URL https://ieeexplore.ieee.org/document/10378364.

APPENDIX

A. Dataset Generation Details

The simulation environment used for data generation and simulation testing is PyBullet [8] (Fig. 9). The object arrangements for the CNABU training datasets were created according to the following procedure: First, we sample the desired occupancy fraction of the shelf floor plan - which we set to be between 30 and 45%. We define two parameters for each object class: its affinity to other classes and its radius of influence. Classes within an object's affinity class and radius of influence have their probability of appearance increased. Further, we define a regularity parameter, ρ for the generation, from 0 to 1. When this parameter is zero, there is no enforced alignment in the shelf and when it is one, there is a higher probability for objects to be placed directly in line with the centroid of previously placed objects, emulating more orderly arrangements by increasing the probability of sampling areas directly in front or behind already placed objects.

We begin sampling by placing a fine grid over the floor space of the scene. Then, until the desired shelf occupancy is reached (or a total number of iterations is reached), we sample a point in this grid that is not yet occupied by another object - taking into consideration the altered probabilities of occupancy by ρ. Through the use of Minkowski differences between the object shapes and the free space, we determine the placeable area for the centroid of each object class within the current arrangement and, for each object that is placeable in the selected point, we compute its sampling probability conditioned on the affinities of the objects whose area of influence contains the sampled point. We then sample an object class according to that distribution and randomly sample an angle for the object, between 0° and 180° and place the object in that orientation. To stimulate the presence of occlusions, we set the base probability of larger objects to be slightly higher - and set large objects to have affinity for smaller objects. For dataset generation, we leave ρ at 0. The ground truth map for each scenario is collected by removing the top shelf, taking a series of dense top-down images of the scene and mapping them with traditional metric-semantic grid mapping [28]. For the pushing dataset, the scenario sampling is identical, except we also sample a random push following Sec. IV-D.

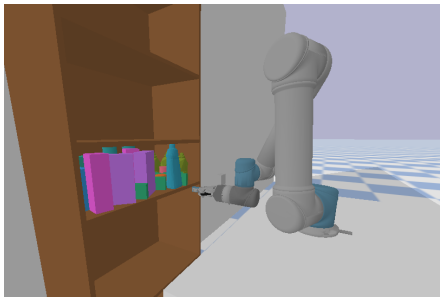
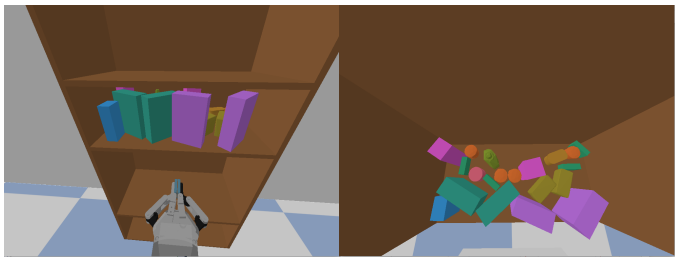
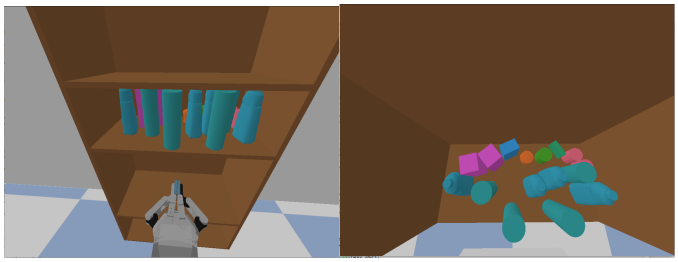


Fig. 9: Simulation environment configuration example of different YCB objects in a confined shelf scenario. The UR5 is equipped with an Robotiq parallel-jaw gripper



(a) High Occupancy. sample - frontal (b) High Occupancy sample - top

Fig. 10: Frontal and (privileged) top-down views of a scene from the highly occluded set



(a) Low Occupancy sample - frontal (b) Low Occupancy sample - top

Fig. 11: Frontal and (privileged) top-down views of a scene from the Slightly Occluded set (right).

Figures 10 and 11 show example scenes from the High Occupancy and Low Occupancy sets. Note how both scenes have occluded objects which are hard to see from any viewpoint - but the HOS scene has more occluded objects - and a much more challenging manipulation environment to uncover the occlusions.

B. CNABU Implementation Details

Each CNABU implements a preprocessing step to encode actions and observations in a representation aligned to the map grid. For the observation CNABU $\sigma_o(\boldsymbol{\lambda}, a_t, o_t)$, we first project the observation occupancies and semantics into an aligned map space using the chosen viewpoint (Free Cell Obs., Occ. Cell Obs., and Semantic Obs. in Fig. 3). We then learn $\sigma_o \equiv \sigma_o(\boldsymbol{\lambda}, \text{Project}(a_t, o_t))$. For the manipulation CNABU, the robot trajectory a_t is projected into an aligned map space that approximates the robot's swept volume. To calculate the robot's swept volume, we approximate the robot's end-effector and last 2 links with simple convex shapes (triangular prism and boxes) and subdivide the robot's path, marking all voxels within those primitives as being part of the swept volume, which is then encoded in a binary 3D voxel grid (swept volume in Fig. 3). Additionally, the trajectory start and end points are encoded in 2D binary masks (Start Point and End Point maps in Fig. 3). Ultimately, we learn $\sigma_m \equiv \sigma_m(\boldsymbol{\lambda}, \text{RobotOccupancy}(a_t(t_s)), \text{RobotOccupancy}(a_t(t_e)))$.

We use network architectures similar to Georgakis et al. [12], with the exception that the output heads are set to be posterior networks. Our network structure can be seen in Fig. 3 The occupancy head in both architectures is a 1x1 convolution, while the differences head in the push

TABLE II: Summary of features of all considered baselines

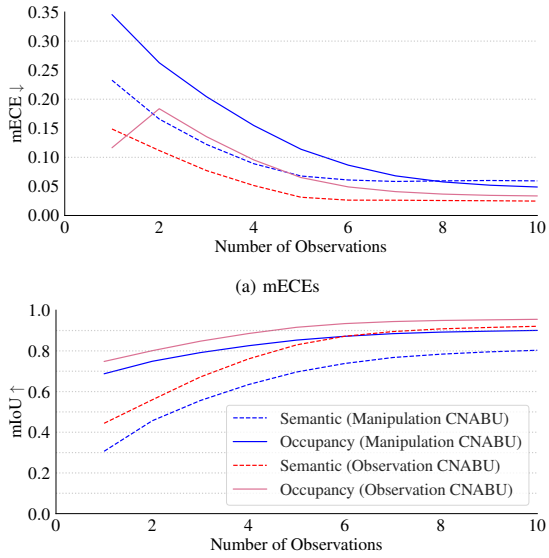
Baseline Name	Use σ_o	VPP	Use σ_m	Push	Action Selection
Ours	yes	yes	yes	yes	Sec. IV-C
Ours w/o Pushing	yes	yes	no	no	no
Random + Observation CNABU	yes	no	no	no	no
Random	no	no	no	no	no
Random Push Every 5^{th}	yes	yes	yes	yes	Random every 5 steps
Random Push Every 5^{th}, No Push CNABU	yes	yes	no	yes	Random every 5 Steps
Random View Random Push Every 5^{th}	yes	no	no	yes	Random every 5 steps
Saturation Pushing	yes	yes	yes	yes	Random upon VPP saturation
Dengler et al. [11]	no	yes, RL-Based	no	yes	Push classifier network upon saturation

prediction network is a series of Convolutional + ReLU + BatchNorm layers followed by a 1x1 convolution. The semantic head is similar to the differences head. The 2D projection block merely takes a horizontal slice of the occupancy tensor at a fixed height, in our case, 3cm above the shelf. Their losses and training are described in Sec. IV-G. We used BEVFusion's approach of feeding the voxel heights as additional channels to 2D Res-UNets for processing the voxel grids as inputs [27, 24].

The networks are trained using backpropagation in PyTorch [32], with grid search-optimized learning rates and ADAM optimizer, as well as early stopping based on the validation loss. The dataset for training σ_o consists of 30.000 randomly sampled scenes, while the dataset for training σ_m consists of 11.700 pushes. Both datasets were split into train, validation and test splits at a ratio of 0.8:0.1:0.1. Dataset generation details are discussed in Sec. A of the appendix. For added robustness in real-world scenarios, we augment the simulation data with salt-and-pepper noise, random rotations and translations and add Gaussian noise to the depth images.

C. Individual CNABU Performance Evaluation

To evaluate the performance of the trained CNABUs, we use the unseen test set of the dataset used for their training. To evaluate the observation CNABU σ_o, we select, for each of the scenes, 10 viewpoints to observe at random and obtain the beliefs at each time step, comparing them against the ground truth map. For evaluating the performance of the manipulation CNABU. σ_m, we also choose 10 viewpoints at random and obtain the pre-manipulation beliefs at every time step, and then obtain the predicted belief after the manipulation is executed. This evaluates the performance of the manipulation CNABU at different reconstruction steps. For each network, we report their mean Intersection over Union and their mean Expected Calibration Error (mECE) [28] for both the semantic and occupancy beliefs in Fig. 12. The mIoU serves as a measure of the correctness of the predicitons, while the mECE measures the confidence calibration of these predictions, *i.e.*, how well the predicted confidences align with actual network performance. Note how after few observations, both networks achieve off-the-shelf reasonable calibration and accuracy, which improves as more views are obtained. It is worth noting that the calibration of the manipulation CNABU with very few observations is low because there is no guarantee that one or two random views would have captured the area being manipulated when predicting the belief after the push.



(a) mECEs

(b) mIoUs

Fig. 12: mECEs and mIoUs for the CNABUs on their respective test sets.

D. Validating Visual Information Gain

Our proposed planner relies on the assumption that the Volumetric Occlusion-aware Information Gain heuristic, which was developed to estimate information gain in traditional occupancy grid maps, translates well to maps that are no longer fully independent, but predicted via CNABUs. We validate this assumption with the following experiment. Consider the pure Viewpoint Planning task, i.e., we must survey the environment without manipulating it, which is a submodular optimization. Consider now a greedy clairvoyant oracle policy, which, at every time step, has access to all possible observations that could be taken and selects the one that leads to the largest information gain. More formally, this policy is defined by:

$$v_t^{\text{clairvoyant}} = \arg\max_{v \in \mathbb{V}} \left[H(\Phi_{t-1}) - H\left(\sigma_o\left(\Phi_{t-1}, v\right)\right) \right] \quad (8)$$

where H denotes the entropy of the maps. We compare our agent without pushing to this privileged information agent in the high-occlusion set of scenes and report the resulting mean map occupancy entropies in Fig. 13. Note how our method which relies on the Occlusion-aware information gain heuristic closely tracks the performance of the privileged

clairvoyant oracle in terms of map entropy reduction. While more extensive studies are encouraged, this suggests that this heuristic still performs well, even when the maps are being predicted via CNABUs.

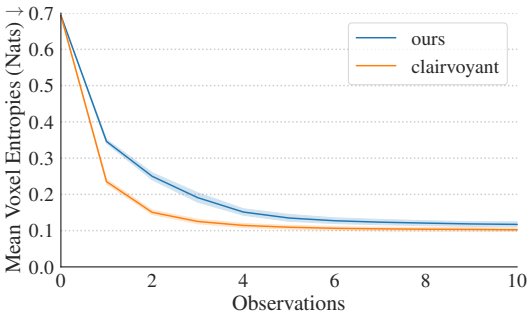


Fig. 13: Evaluation of the occlusion-aware information gain heuristic against a privileged clairvoyant view selection policy

E. Prompts used for Semantic Segmentation

To perform semantic segmentation of the SAM2 masks, we use the embeddings of the following text prompts to classify the segments:

- tomato can -"tomato or kidney beans round tin can"
- chewing gum -"small round chewing gum box"
- spam: "potted meat tin can, spam"
- ohio cookies - "ohio cookie box in purple cardboard carton cookies",
- mustard bottle - "yellow frenchy's mustard bottle",
- coffee can - "maxwell house coffee can with blue wrapper",
- gelatin box - "light pink gelatin box",
- cheez-it box - "cheezeit cracker box in dark red color",
- pringles can - "pringles chips tube cyllinder red or green, in red color or green color bottles with transparent lid",
- glass cleaner - "glass cleaner spray plastic bottle",
- baking powder - "koop mans baking powder box",
- pasta box - "Big blue carton of pasta collezione"
- Cornflakes - "kölln schoko cronflakes in yellow brown cardboard box cereal"
- milk carton - "milk box tetrapak blue label voll milch",
- shelf - "wooden shelf board","cream colored wooden shelf of light cream color"
- black - "just black"

In this case, both "shelf" and "black" were used as synonymous of the background class, capturing different failure cases of SAM2 segmentation.

F. Summary of baseline features

We summarize the considered baselines in Tab. II to facilitate comparing their features.

G. Physical Experiments Object Set

Fig. 14 shows the objects which were used to create the test scenes during our real world experiments. Note how there are multiple shapes for objects of the same class, like pringles cans, milk cartons and cans. They also differ from the geometries used during training.

Fig. 14: The set of objects used during the real world experiments