Semantically Informed MPC for Context-Aware Robot Exploration

Yash Goel*[1,2] Narunas Vaskevicius*[2] Luigi Palmieri[2] Nived Chebrolu[3] Kai O. Arras[2] Cyrill Stachniss[4]

***Abstract*— We investigate the task of object goal navigation in unknown environments where a target object is given as a semantic label (e.g. find a couch). This task is challenging as it requires the robot to consider the semantic context in diverse settings (e.g. TVs are often nearby couches). Most of the prior work tackles this problem under the assumption of a discrete action policy whereas we present an approach with continuous control which brings it closer to real world applications. In this paper, we use information-theoretic model predictive control on dense cost maps to bring object goal navigation closer to real robots with kinodynamic constraints. We propose a deep neural network framework to learn cost maps that encode semantic context and guide the robot towards the target object. We also present a novel way of fusing mid-level visual representations in our architecture to provide additional semantic cues for cost map prediction. The experiments show that our method leads to more efficient and accurate goal navigation with higher quality paths than the reported baselines. The results also indicate the importance of mid-level representations for navigation by improving the success rate by 8 percentage points.**

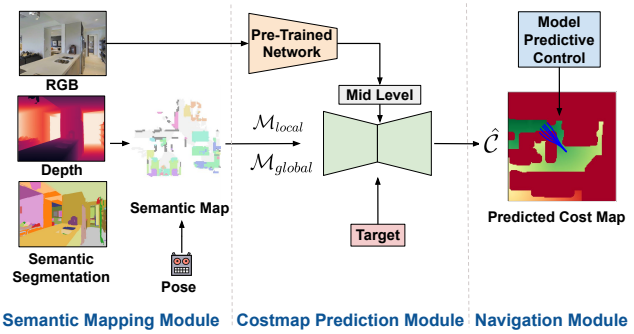


Fig. 1: Our exploration framework for object navigation task is composed of three modules: *semantic mapping module* which constructs the map of the environment as the robot explores. The *cost map prediction module* which predicts the cost map for navigation based on the input semantic map, mid-level representation and target object. Finally, the *navigation module* generates the optimal control using a sampling-based MPC.

I. Introduction

Equipping a robot with semantics-aware navigation skills is essential for intelligent and efficient behavior in complex human-made environments. For example in a house, a robot tasked with finding a couch should be able to draw conclusion that if it is near a TV then the couch should be nearby - since they generally tend to be in the same space (*living room*). It is important to learn this kind of semantic information to make better decisions about where to go and how to get there. In this work, we investigate how to leverage this contextual information to efficiently find target objects while exploring unknown environments.

Reliable and accurate semantic robot navigation is still an open research question [1]. Traditional approaches use semantic knowledge for building graphs [2] or try to navigate to rooms [3] using various planning approaches, however those approaches tend to rely on hand-defined features and representations. The latter being built on top of various perception algorithms like object detection or semantic segmentation.

Recently with the surge of deep learning for computer vision and reinforcement learning various new methods have been proposed to tackle this problem. End-to-end control learning [4], [5], hybrid approaches combining traditional planning with RL [6], [7], among others have been proposed. Many of these works use reinforcement learning and tend to use only a limited discrete action space. They focus on policies with simple actions like (left, right and straight), thus resulting in non-smooth and possibly dynamically infeasible robot behaviors.

To achieve efficient and dynamically feasible *object goal navigation*, i.e. looking for and reaching a defined semantic target, we propose a technique that combines model-based continuous control approach with perception module that exploits semantic information and mid-level feature representations.

We summarize our main contributions as follows:

(i) We present a semantically informed Model Predictive Control (MPC) approach for efficient context-aware robot exploration. The approach combines a sampling-based model predictive control technique with cost map predictions based on deep neural network, that implicitly considers semantic information about objects and places in the environment.

(ii) We propose a U-Net based architecture for dense cost map prediction under partial observability. Further we explore the use of egocentric mid-level visual representations in the network architecture. We present a novel approach of fusing these features to our network in an robot-orientation-aware way.

(iii) The approach is shown to outperform a set of

[1]Y. Goel did this work while with the University of Bonn, Germany, and Robert Bosch GmbH. {s7yagoel}@uni-bonn.de.

[2]L. Palmieri, N. Vaskevicius, K.O. Arras are with Robert Bosch GmbH, Corporate Research, Stuttgart, Germany. {luigi.palmieri, narunas.vaskevicius, kaioliver.arras}@de.bosch.com.

[3]N. Chebrolu is with with the Dynamic Robot Systems Group, University of Oxford, UK. {nived}@robots.ox.ac.

[1,4]C. Stachniss is with the University of Bonn, Germany, with the Department of Engineering Science at the University of Oxford, UK, and with the Lamarr Institute for Machine Learning and Artificial Intelligence, Germany. {cyrill.stachniss}@igg.uni-bonn.de.

*Denotes equal contribution.

This work was partly supported by the EU Horizon 2020 research and innovation program under grant agreement No. 101017274 (DARKO).

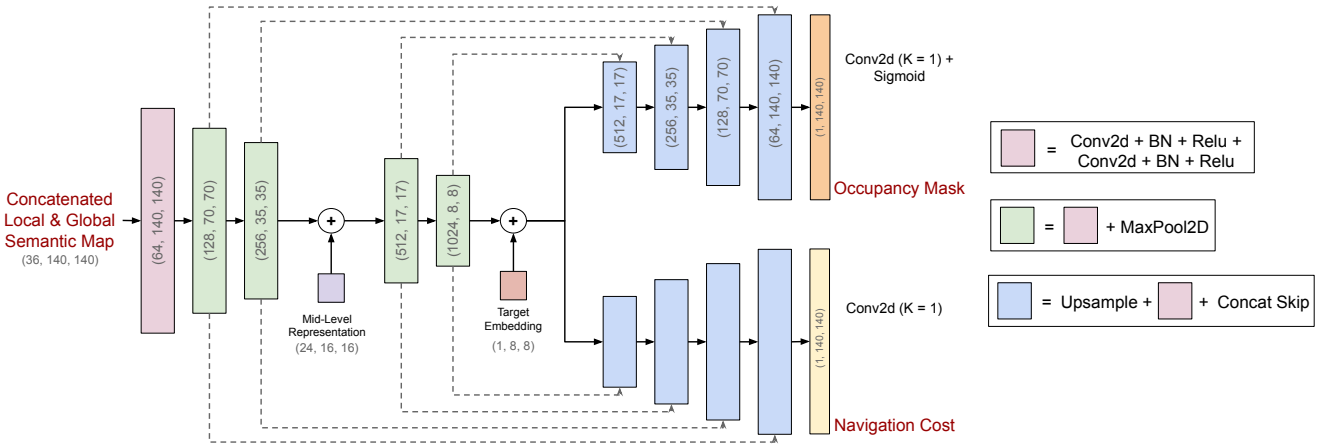


Fig. 2: Our detailed network architecture for cost map prediction. Dashed arrows denote skip connections. Kernel size K as 3, stride S as 1 and padding P as 1 are taken unless specified otherwise. The mid-level representations are fused according to orientation bin as shown in Fig. 3. In the end, the occupancy mask and navigation cost are fused to get the final cost map prediction.

baselines in terms of efficiency and final path quality. Our experimental evaluation shows that mid-level representations significantly (by 8 percentage points in success rate) improve the navigation performance.

II. Related Work

Several prior works have tried to use semantics for navigating in cluttered and changing environments. Two main paradigms are map-based approaches and visual/perception-based ones.

1) Map-Based Prediction: Cost map prediction for navigation has been widely used in the literature. Fan et al. [8] use a deep learning-based approach to predict traversability costs for unknown and unstructured environments. Drews et al. [9] predict cost maps from input videos using CNNs, which are then used for online trajectory optimization with MPC. Qi et al. [10] learn to predict a spatial affordance map through active self-supervised experience. Chaplot et al. [11] use transformers to predict distances by exploiting their property of learning long-distance relationships instead of local convolutional features. Zhu et al. [12] predict navigation costs at the frontiers of robot map and using those for semantic-aware navigation. Unlike our approach, they do not predict the cost map for the whole map but select a waypoint to follow from the frontier. Instead of predicting only at the frontiers we provide dense predictions: the latter is a more natural and common representation for downstream tasks such as planning and control. Further, their approach is constrained to a discrete action space whereas our approach is based on continuous control. The work by Ramakrishnan et al. [13] uses occupancy prediction as a way to learn the priors and help the robot in its navigation. Our work predicts an occupancy mask and navigation cost from an incomplete map (i.e. more challenging because the structure of the environment is not known a priori).

2) Visual Navigation: Ko et al. [14] use depth image to determine free space and choose the direction to go using a topological map (i.e. built for target navigation). Recent works by Morad et al. [15] and Chen et al. [4] directly learn policies for navigation to goal. Gupta et al. [7] present a differentiable mapper and planner for generating discrete actions to navigate to visual targets, which are trained end to end to ensure the mapper learns the operation best suitable for the planning module. Chaplot et al. [6] build top-down semantic maps of the world which is fed to a reinforcement learning algorithm for exploration. Contrary to our approach, their approach uses only a set of discrete actions.

The works [16], [17] use mid-level representations for navigating in unknown environments. We build on their results and extend them to continuous control settings. Other methods use optimal control for navigation but not in object goal navigation setting. Information theoretic model predictive control (IT-MPC) by Williams et al. [18] is used by Drews et al. [9] to learn cost maps for aggressively driving a miniature car around a loop circuit. Our work is similar to [9] in the sense that we generate cost maps for navigation which are further used by IT-MPC for fulfilling an object goal navigation task. Kusumoto et al. [19] learn obstacle-aware sampling distributions for guiding the IT-MPC exploration, thus improving the overall task efficiency of the approach, but do not learn a cost-function for helping the robot during the navigation. From point goal navigation perspective, Bansal et al. [20] use model predictive control to move towards a waypoint generated by a learning-based module using only RGB and localization. Our approach predicts a cost map that implicitly infers the global goal to achieve.

III. Our Approach

A fundamental subproblem of embodied intelligence is the *object goal navigation task* [6], [21], a semantically aware variant of robot exploration: given an initial position for the robot $\mathcal{A}$ (i.e., hereinafter also called agent) and a target object category $\mathcal{T} \in \mathbb{T}$, the agent needs to reach the defined target (e.g., a target category like *bed*). The agent does not know a-priori the environment, it needs to

explore it and to understand potential semantic relationships between objects and places, and thus use those for efficiently fulfilling the given task. The agent has access to only sensor observations ($\mathcal{O}_t$, e.g., ego-centric RGB, depth and semantic segmentation) and current state, $\mathbf{x}_t \in \mathcal{X}$, $\mathcal{X}$ being the space of all possible robot states.

To solve the previously defined task we introduce our semantically informed MPC approach detailed in Fig. 1. The whole framework can be divided into three major components. First component is *semantic mapping*, which is discussed in Sec. III-A, builds the semantic map as the robot observes the environment. The output from this module is send to *cost map prediction network*, which is discussed in the Sec. III-B. We use the predicted cost map in the *navigation* module as detailed in the Sec. III-D.

A. Semantic Map Generation

We follow the approach of Chaplot et al. [6] to construct the semantic map $\mathcal{M}$ of the environment. Overall, it contains the information of obstacles, explored area and top-down semantic information of each grid cell in the map. A given semantically segmented point cloud (e.g. obtained using Mask R-CNN [22] and a depth image) is converted to top-down 2D semantic map where each cell has a semantic label with different probabilities for each class. For the K number of semantic classes we have the semantic map of size (K, N, N) where N is the size of local spatial region that we see in each view. We use 16 semantic classes which is a superset of our target classes to represent our semantic map. We further concatenate this map with obstacle mask and explored mask to finally get map $\mathcal{M}_t$ of size (C, N, N) where $C = 2 + K$ for time t. The global semantic map, $\mathcal{M}_{global}$ is built by fusing these maps $\mathcal{M}_t$ over time using the pose information.

B. Cost Map Prediction

Given as input the semantic map and mid-level representations [17], we aim to predict navigation cost based on our given target label $\mathcal{T}$. The cost implicitly encodes contextual relationships between places and objects in the explored environment. The network architecture is inspired by the U-Net architecture [23]. The whole module architecture can be seen in the Fig. 2. We discuss the various components in this section.

Semantic Map. We take two kinds of input from our semantic map built in III-A - namely local semantic map, $\mathcal{M}_{local}$ and global semantic map, $\mathcal{M}_{global}$. The local semantic map is of the size (C, H, W) where H and W define the spatial size around the robot for which we do cost map prediction. The global map is reduced to spatial size of local map using average pooling. They are concatenated across the channel before being given as an input to the network.

Mid-Level Representation. Mid-level representations are features generated from encoders which have been trained for different downstream tasks like *semantic segmentation*, *denoising*, *curvatures*, *keypoints*, etc. They have shown to be quite effective in RL setting for various downstream tasks [16], [17]: they improve generalizability and also performance of an RL agent. In our approach, we adopt mid-level representations and show how they can be beneficial for predicting context-aware dense cost maps.

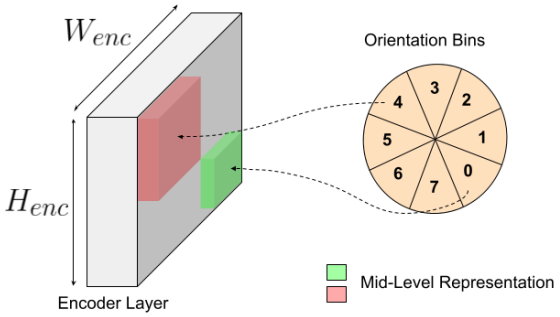


Fig. 3: Orientation of the robot decides the orientation bin in which the mid-level feature falls. Each bin is of the size 45° so that we have a total of 8 bins to cover all the orientation possibilities. For example, the agent looking towards the north will have the bin 3. Two examples illustrated above show the placement of mid-level features corresponding to the agent facing north-west (red color and bin 4) and south-east (green color and bin 0) directions.

The mid-level representations are extracted from the input RGB image using a pre-trained network as used by Sax et al. [17]. We combine encoded mid-level features from three downstream tasks. We use *semantic segmentation* and *object classification* to help learn about object goal and semantic contexts of the world. Apart from this, we use *depth prediction* since we are learning distance dependent cost map. These representations are concatenated to the encoded map features after first two encoder layers as shown in Fig. 1. This concatenation is done according to the orientation of the robot to form *oriented mid-level representation* as shown in Fig. 3. The corresponding bin region is set to the mid-level feature while the rest of layer is set to 0 to form the *oriented mid-level representation* which is concatenated to the encoded map features. Using this binning technique we make these mid-level representations robot orientation aware and associate with the corresponding semantic input map region.

Target Embedding. We encode the target object category that the agent has to reach. To do this, we create an embedding using the index of the target category. The embedding of size M, where $M = 64$, is transformed to the spatial size of the encoded latent feature (H_{enc}, W_{enc}). It is then concatenated to the latent feature along the channel dimension.

Output. We observed that casting the cost map prediction as a multi-task learning problem leads to better results. Therefore, our network consists of two decoder branches. One predicts the occupancy map $\hat{\mathcal{C}}^{occ}$ and the other decoder branch predicts navigation cost $\hat{\mathcal{C}}^{nav}$. We combine $\hat{\mathcal{C}}^{occ}$ and $\hat{\mathcal{C}}^{nav}$ to get the final cost map prediction $\hat{\mathcal{C}}$. This is done by using an occupancy threshold θ_{occ} to create a binary occupancy mask from $\hat{\mathcal{C}}^{occ}$. The predicted navigation cost $\hat{\mathcal{C}}^{nav}$ is only used for free space from the binary mask while

the occupied space is set to high penalty.

C. Loss Function

In this section we describe our multi-term loss function that we formulated to train the network for the cost map prediction. In the following equations we denote the ground truth occupancy mask by $\mathcal{C}^{occ}$ and the ground truth navigation cost by $\mathcal{C}^{nav}$. The dataset generation technique for ground truth has been discussed in the Sec. IV-A.

Occupancy Loss. We use a binary cross entropy loss over the local map region of size (H, W) to learn the occupancy probabilities $\hat{c}^{occ}_{i,j}$ of a map cell (i, j):

$$\mathcal{L}_{occ} = \frac{1}{HW} \sum_{i,j} -c^{occ}_{i,j} \log\left(\hat{c}^{occ}_{i,j}\right) - \left(1 - c^{occ}_{i,j}\right) \log\left(1 - \hat{c}^{occ}_{i,j}\right), \tag{1}$$

where $c^{occ}_{i,j}$ is 1 in case of obstacle and 0 in case of free space. This loss term is used only for the branch predicting the occupancy map.

Cost Map Loss. To learn the cost map prediction, we regress the navigation cost using the L1 norm, averaged over all valid positions (i.e. navigable area):

$$\mathcal{L}_{cost} = \frac{1}{HW} \sum_{i,j} \left\| \left(c^{nav}_{i,j} - \hat{c}^{nav}_{i,j}\right) \left(1 - c^{occ}_{i,j}\right) \right\|_1, \tag{2}$$

where $c^{nav}_{i,j}$ is the normalized ground truth navigation cost and $\hat{c}^{nav}_{i,j}$ is the predicted navigation cost.

Gradient Direction Loss. We introduced this term to make the navigation cost smooth and consistent with the local ground truth gradients. Similar to cost map loss, we only calculate this loss over navigable area of the map:

$$\mathcal{L}_{dir} = \frac{1}{HW} \sum_{i,j} \left(1 - \frac{\mathbf{g}_{i,j} \cdot \hat{\mathbf{g}}_{i,j}}{|\mathbf{g}_{i,j}| \cdot |\hat{\mathbf{g}}_{i,j}|}\right) \left(1 - c^{occ}_{i,j}\right), \tag{3}$$

where $\mathbf{g}$ is the gradient for ground truth cost map and $\hat{\mathbf{g}}$ is the gradient for predicted cost map,

$$\mathbf{g}_{i,j} = \left(\frac{\delta \mathcal{C}^{nav}}{\delta x}, \frac{\delta \mathcal{C}^{nav}}{\delta y}\right)_{i,j}$$
$$\hat{\mathbf{g}}_{i,j} = \left(\frac{\delta \hat{\mathcal{C}}^{nav}}{\delta x}, \frac{\delta \hat{\mathcal{C}}^{nav}}{\delta y}\right)_{i,j}.$$

In our experiments we observed that the addition of this term leads to significantly smoother cost maps, which is important for the downstream navigation task.

The total loss then becomes a combination of all these losses,

$$\mathcal{L}_{total} = \alpha_{occ}\mathcal{L}_{occ} + \alpha_{cost}\mathcal{L}_{cost} + \alpha_{dir}\mathcal{L}_{dir}, \tag{4}$$

with the empirically selected weights $\alpha_{occ} = 1.0$, $\alpha_{cost} = 1.5$ and $\alpha_{dir} = 1.0$.

D. MPC-based Navigation

The usage of Model Predictive Control has largely replaced the typical cascades of loosely coupled planning and control layers in robot navigation. Here we adopt a formulation of MPC which is particularly suited to tightly integrate the dense cost functions.

As the learned cost functions are expected to be highly nonlinear and classical numerical optimization methods may fatigue with complex cost landscapes, we use a sampling-based approach. Specifically, we use IT-MPC (Infomation Theoretic Model Predictive Control [18]) for the robot to find optimal control sequences. We start by having the initial control sequence $U = \{\mathbf{u}_0, \mathbf{u}_1, \dots \mathbf{u}_{H-1}\}$ where H is the horizon of the IT-MPC. Each control in the sequence is then perturbed for K samples to generate noisy control, $\tilde{U}_k = U + \mathcal{E}_k$ where $\mathcal{E}_k = \{\epsilon_0, \epsilon_1, \dots \epsilon_{H-1}\}$. Every noise ϵ_t is sampled from a normal distribution $\mathcal{N}(\mu, \sigma)$ using $\mu = 0$ and empirically selected $\sigma = 0.35$.

For each sample $\tilde{U}_k$, we generate the semantically-enriched cost $\mathcal{S}_k$ using the predicted cost map and the control effort.

$$\mathcal{S}_k = \sum_{t=0}^{H-1} \left(\hat{\mathcal{C}}_t(\mathbf{x}_t) + \mathbf{u}_t^T Q \mathbf{u}_t\right), \tag{5}$$

where $\mathbf{x}_t = \{x_t, y_t, \theta_t\}$ is the robot pose and $Q \geq 0$ is the control effort matrix. The robot pose is sampled using a constant velocity differential drive model using the perturbed linear and angular velocity ($\mathbf{u}_t = \{v_t, \omega_t\}$ where v_t and ω_t).

The cost is then used to generate weights w_k of the importance sampling step that obtains the optimal control sequence to execute: $\beta = \min_k S\left(\mathcal{E}^k\right), \eta = \sum_{k=0}^{K-1} \exp\left(-\frac{1}{\lambda}\left(S\left(\mathcal{E}^k\right) - \beta\right)\right), w_k = \frac{1}{\eta} \exp\left(-\frac{1}{\lambda}\left(S\left(\mathcal{E}^k\right) - \beta\right)\right)$. Finally, the control sequence $\mathbf{u}_t$ is updated using the weights and control noise.

$$\mathbf{u}_{t+1} = \mathbf{u}_t + \sum_{k=1}^{K} w_k \epsilon_t^k. \tag{6}$$

Hence, we get the updated control and we apply the first control $\mathbf{u}_0$ to the agent.

Goal Reacher. During the exploration, if the target object category $\mathcal{T}$ is observed in the local top-down semantic map $\mathcal{M}_{local,t}$ then the corresponding map cells define the goal mask $\mathcal{M}_{goal,t}$. To avoid false positive cells in goal mask, we remove small regions from the mask. Then using the remaining goal mask $\mathcal{M}_{goal,t}$ and local occupancy map $\mathcal{M}^{occ}_{local,t}$ we generate cost map for navigation using Fast Marching Method (FMM) [24] by setting goal mask region as 0 value:

$$\mathcal{C}^{goal}_t = \mathrm{FMM}\left(\mathcal{M}^{occ}_{local,t}, \mathcal{M}_{goal,t}\right). \tag{7}$$

The agent then drives using this cost map. It declares `done` when the cost map value is less than or equal to the cost threshold θ_{cost} i.e. $\mathcal{C}^{goal}_t \leq \theta_{cost}$. In our experiments we used $\theta_{cost} = 0.2$. If the goal turns out to be unreachable due to previously unobserved obstacles then our approach leaves the goal reaching mode and resorts to the predicted cost map to continue the exploration.

IV. Experimental Setup

We perform experiments in the real-world indoor environments provided by a large-scale RGB-D dataset Matterport3D (MP3D) [25]. We use a physics-enabled 3D simulator Habitat [26] to navigate the agent in these environments. To train our cost map prediction network we generate a dataset as described in Sec. IV-A. We describe the evaluation setup and the metrics for cost map prediction and navigation in Sec. IV-B and Sec. IV-C respectively. Finally, Sec. IV-D provides important implementation details.

A. Cost Map Prediction Dataset

Our interest is in house-like environments containing objects, such as a couch, a bed, a table, a chair, a plant, etc. Therefore, as a first step, we filter out environments from MP3D dataset which do not contain relevant semantic information e.g. large halls or churches. In addition, we omit the houses containing multiple incorrect object labels, which can impair the training process. The remaining 48 houses form our dataset are divided into the train, validation and test splits consisting of 36, 4 and 8 houses respectively.

For each house, we sample multiple starting points and randomly select a goal from the set of goals we are considering. For each floor where the robot is spawned, we get the ground truth top-down semantic map as defined in [27] which is then used to generate the goal map based on the target object. Combining this goal map with the occupancy map from the Habitat simulator, we generate the global ground truth cost map of distances.

We use an MPC-based agent with the ground truth cost maps to reach the goal while we collect the dataset. We record samples at every fourth timestep to reduce redundancy. The number of trajectories taken in a house depend on the size of the house to avoid repetition. This was selected manually for each house upon inspection. The complete generated dataset contains 171412, 16209 and 41949 samples in the train, val, and test splits respectively.

Each training sample consist of i.) local semantic map ($\mathcal{M}_{local}$) of size 140×140, ii.) global semantic map ($\mathcal{M}_{global}$) of size 420×420 to help capture global context, iii.) ground truth cost map ($\mathcal{C}$) composed of distances to the goal using FMM [24] and occupancy map, and, iv.) egocentric RGB for computing the mid-level visual representations [17]. We also save the orientation of the robot along with the image.

B. Evaluation Setup for Cost Map Prediction

We evaluate both occupancy and cost map prediction for our approach. The predictions are evaluated on the test split of the generated dataset (Sec. IV-A). The occupancy prediction uses classification metrics of mean F1 score (mF1) and mean Intersection over Union (mIOU) averaged over both free space and occupied space classes. We also report mean pixel accuracy (MPA) for occupancy prediction. For navigation cost prediction, we report average Action Prediction (aAP) which determines the accuracy of picking the right local policy normalised by navigable area. aAP_5 determines per-pixel accuracy of picking the correct action based on the lowest cost from 4 basic directions and being stationary. Similarly, aAP_9 measures the same but with 8 neighbours and the robot position. aAP_9 gives us a more accurate resolution as the agent can move in diagonal directions as well.

C. Evaluation Setup for Object Goal Navigation

For navigation performance we ran the agent on different houses from the test split in the Habitat simulator. There are a total of 8 test houses - for each house we sample 40 random starting positions for the robot along with a random target object to reach. We gather various metrics related to success of reaching the target object like success of reaching the goal, SPL [28], - which weighs the success according to the path length determining its efficiency and DTS [6] i.e. Distance to Success, which measures how far is the agent from the success distance, which in our experiments was set to 1 m. We also measure the smoothness of the final robot trajectory using average acceleration and jerk. We run each experiment for 500 timesteps which is equivalent to 50 s. The target object is selected from any of the target category list that we consider: *bed*, *chair*, or *couch*.

D. Implementation Details

The cost map prediction network was implemented in PyTorch. During training, we applied data augmentation by randomly rotating (with a probability of 0.15) the input semantic maps and the target cost maps by 90°, 180° or 270°. We used the SGD optimizer and a constant weight decay factor of 0.01. In all experiments, the learning rate followed the cosine decay schedule with a warmup phase of 25 epochs with a peak and terminal learning rates being $15e$-5 and $1e$-5 respectively. All cost map prediction models were trained for 200 epochs. For the MPC we used a horizon $H = 50$.

V. Experimental Results

A. Quantitative Results

In this section, we analyze the quantitative results of our approach for context-aware robot exploration using semantically-informed MPC.

Costmap Prediction. For occupancy mask prediction, we get an MPA of 78.7%, mF1 of 75.8% and mIOU of 62.8%. We observe that the F1 score of occupied region is 80.7% and that of free region is 71%. This shows that our occupancy mask prediction approach is inclined to predict the occupancy class better than the free space class. Similar trend was seen for IOU. For occupied region, it is 69.1% and for free region is 56.7%. We get scores for aAP_5 and aAP_9 as 37.5% and 33.4% respectively. An example of our prediction can be seen in Fig. 4. We further ablate costmap prediction based on semantic input in Sec. V-C.

Navigation. We compare our approach for object goal navigation with the following continuous action space agents:

1. *GT Agent*: This approach has access to ground truth cost map which is then used for navigation by the MPC.

Approach	SR ↑	SPL ↑	DTS(m) ↓	Timesteps ↓	Acc(ms^{-2}, $rads^{-2}$) ↓	Jerk(ms^{-3}, $rads^{-3}$) ↓
GT Agent	1.000	0.922	0.219	145	$[0.17, 2.37]$	$[2.37, 35.00]$
Privil. Random	0.282	0.206	7.136	394	$[1.57, 8.64]$	$[28.00, 156.10]$
FBE + MPC	0.437	0.277	5.137	336	$[0.21, 5.59]$	$[3.36, 101.82]$
SemExp [6] + MPC	0.349	0.273	6.174	**292**	**[0.19, 1.57]**	**[2.95, 28.12]**
Our Approach	**0.520**	**0.390**	**3.594**	304	$[0.66, 7.26]$	$[11.79, 128.16]$

TABLE I: Navigation performance comparison. GT cost map provides an indication of the best possible metrics.

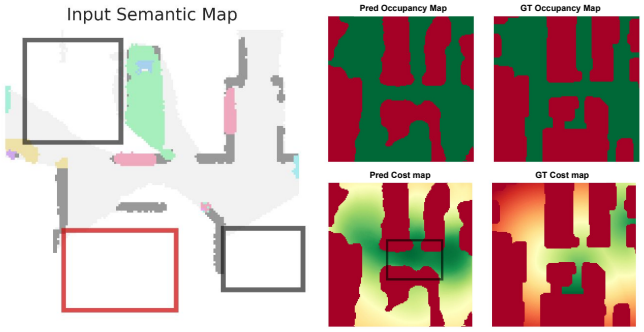


Fig. 4: The left image shows the input semantic map where, white is unexplored area, gray is free space and the rest is other semantic regions. The predicted occupancy map on the top right matches well with the ground-truth occupancy map in the black box regions. Whereas for the red box, our method is extrapolating free space. Here red is occupied and green is free space. For the cost maps on the bottom right, the predicted cost is able to capture the relative lower cost in the center of the map. For cost map, *red* to *green* represent *high* to *low* cost.

This agent is useful to understand the upper bound on the performance.

2. *Priviliged Random*: It picks a random value from the allowable set of linear and angular velocity to use as an action, it is made *privileged* by providing the *goal reacher*, which is semantically informed. This helps us to see the improvement in exploration by our agent.

3. *FBE + MPC*: This agent performs frontier-based exploration with our IT-MPC combined. The costmap is generated by propagating the zero cost from the frontiers towards the agent.

4. *SemExp + MPC*: This agent selects semantically informed long term goals (waypoints) similar to the method from [6]. They guide the robot towards the target object. Instead of the deterministic local policy in discrete action space used in [6], the agent relies on our IT-MPC to approach the waypoints. This is done to enable continuous control of this agent and make it comparable to our approach.

Tab. I shows that our agent outperforms all the baseline methods in the main object goal navigation metrics. It improves over the best baseline FBE + MPC by 8 and 11 percentage points respectively in success rate (SR) and success weighted by the path length (SPL). The performance gap is even larger when compared to the adopted method from [6]. In average our approach reaches the goals significantly closer compared to the baselines as evident from DTS metric.

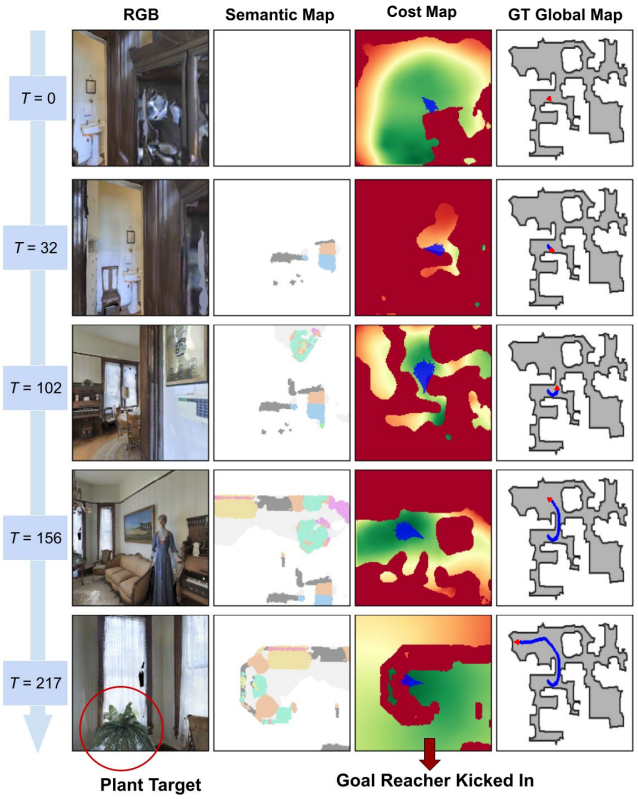


Fig. 5: Progression of agent moving in the house over time. The path over time is shown in *blue* in the GT global map with the *red* arrow showing the orientation of the robot. The target goal in this case is *plant* and we see that the agent is able to navigate to the plant efficiently. We also see in the last timesteps the goal reacher is activated. In the cost maps, the *green* regions show low cost and *red* the high-cost regions. Samples from MPC are also shown in the cost maps in *blue*.

The lower jerk for MPC augmented baselines is caused by smoother costmaps generated using FMM from zero cost regions or waypoints. This indicates a potential direction of improvement for our predicted costmaps. The privileged random and ground-truth agents provide the lower and upper bounds for the navigation performance respectively.

B. Qualitative Results

In this section we present the qualitative results to show how our agent successfully completes the task of object goal navigation on one of the sequences. In this example the target object is a plant showcasing that the approach can work beyond the object categories used for the quantitative

Methods	*Occupancy Prediction*			*Navigation Cost Prediction*		*Object Goal Navigation*		
	MPA(%) ↑	mF1(%) ↑	mIOU(%) ↑	aAP_5(%) ↑	aAP_9(%) ↑	SR ↑	SPL ↑	DTS(m) ↓
Only Semantic Map	**79.1**	76.2	63.2	36.9	33.2	0.437	0.330	4.49
Only Mid-Level	**79.1**	**76.3**	**63.4**	37.2	33.5	0.492	0.371	3.79
Both (Our Approach)	78.7	75.8	62.8	**38.2**	**34.4**	**0.520**	**0.390**	**3.59**

TABLE II: Performance comparison for ablation on semantic input.

evaluation. Fig. 5 shows how the agent progresses towards the goal as it builds the semantic map of the environment. The predicted cost map efficiently guides the agent near the goal area. At timestep 217 the goal reacher (see Sec. III-D) gets activated as the target object (i.e. *plant*) becomes visible in the local semantic map. Finally, the cost map generated by the goal reacher drives the agent to the goal.

C. Ablation Study

In this ablation study, we compare how the different input information affects the costmap prediction and navigation results. We compare three variants: *Only Semantic Map*, which has only semantic map as the input to the cost map prediction module, *Only Mid-Level*, which has input map without semantics along with the mid-level representations and finally, *Our Approach*, with both semantic map and mid-level input.

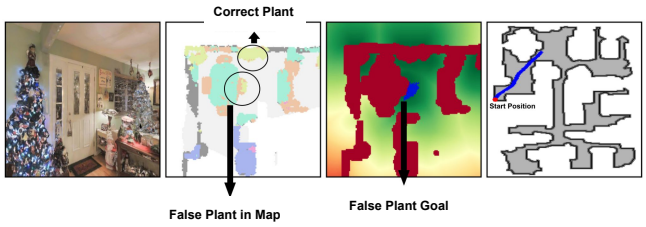
The MPA, mF1 and mIOU metrics reported in Tab. II indicate that occupancy prediction is not influenced by the input variations. The aAP_5 and aAP_9 metrics show that the introduction of mid-level representations improve the navigation cost prediction. This observation is consistent with the results of the downstream object goal navigation task.

In terms of navigation, the *Only Mid-Level* already reaches a success rate of 0.492 while *Only Semantic Map* has a success rate of only 0.437. An addition of mid-level representations greatly improves the success rate of the *Only Semantic Map* by 8 percentage points. This clearly shows that mid-level representations are beneficial for achieving better object goal navigation efficiency. A combination of both semantic representations leads to the best result in terms of success rate, SPL and DTS.

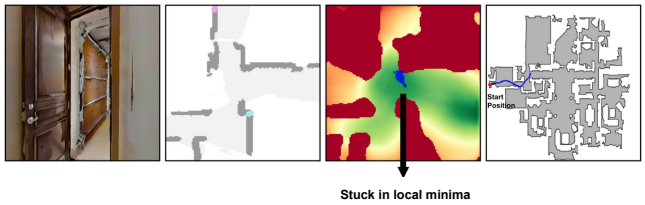
D. Failure Cases

To determine potential directions for the improvement of our approach, we also consider failure cases for which we mainly identified the following reasons:

1) Noisy Semantic Map: Due to the noisy egocentric semantic segmentation, some cells in the constructed semantic map get incorrectly classified. In cases like these e.g. Fig. 6a, it identifies the plant target incorrectly and therefore, declares the end of episode early. This leads to failure in reaching the true goal.
2) Local Minima: Other sometimes observed failure case is that the agent gets stuck in the local minima of the predicted cost. Fig. 6b shows one such case where the robot gets stuck after traversing a considerable distance in search of target goal.



(a) Noisy Semantic Map



(b) Local Minima

Fig. 6: Examples of failure cases observed in our approach.

These failure cases are potential areas of improvement for future work which can increase the exploration performance of our approach.

VI. Conclusion

In this work, we present a semantically-informed MPC approach for context-aware object goal navigation under kinodynamic constraints. Our proposed U-Net based network architecture includes a novel way of fusing mid-level representations which takes into account the orientation of the robot. The chosen sampling-based variant of MPC allows us to develop a lean and tight coupling between semantic modeling components and downstream control. The experiments show that our method achieves more efficient and accurate goal navigation with higher quality paths than the reported baselines. The results also indicate the importance of mid-level representations for navigation by improving the success rate by 8 percentage points. As future work, we wish to perform experiments on real robots and tackle uncertainties as dense cost maps provide an appropriate base for that.

References

[1] J. Crespo, J. C. Castillo, O. M. Mozos, and R. Barber, "Semantic information for robot navigation: A survey," *Applied Sciences*, vol. 10, 2020.

[2] J. G. Rogers and H. I. Christensen, "Robot planning with a semantic map," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.

[3] B. Talbot, O. Lam, R. Schulz, F. Dayoub, B. Upcroft, and G. Wyeth, "Find my office: Navigating real space from semantic descriptions," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2016.

[4] T. Chen, S. Gupta, and A. Gupta, "Learning exploration policies for navigation," in *Proc. of the Int. Conf. on Learning Representations(ICLR)*, 2019.

[5] A. Mousavian, A. Toshev, M. Fiser, J. Kosecká, A. Wahid, and J. Davidson, "Visual representations for semantic target driven navigation," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 8846–8852, 2019.

[6] D. S. Chaplot, D. Gandhi, A. Gupta, and R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," in *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, 2020.

[7] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive Mapping and Planning for Visual Navigation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[8] D. D. Fan, A. Agha-mohammadi, and E. A. Theodorou, "Learning risk-aware costmaps for traversability in challenging environments," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 1, pp. 279–286, 2022.

[9] P. Drews, G. Williams, B. Goldfain, E. Theodorou, and J. Rehg, "Aggressive deep driving: Combining convolutional neural networks and model predictive control," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2017.

[10] W. Qi, R. T. Mullapudi, S. Gupta, and D. Ramanan, "Learning to move with affordance maps," in *Int. Conf. on Learning Representations (ICLR)*, 2020.

[11] D. S. Chaplot, D. Pathak, and J. Malik, "Differentiable spatial planning using transformers," in *Proc. of the Int. Conf. on Machine Learning (ICML)*, 2021.

[12] M. Zhu, B. Zhao, and T. Kong, "Navigating to objects in unseen environments by distance prediction," *ArXiv*, vol. abs/2202.03735, 2022.

[13] S. K. Ramakrishnan, Z. Al-Halah, and K. Grauman, "Occupancy anticipation for efficient exploration and navigation," *ArXiv*, vol. abs/2008.09285, 2020.

[14] D. W. Ko, C. Yi, and I. H. Suh, "Semantic mapping and navigation: A bayesian approach," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[15] S. D. Morad, R. Mecca, R. P. K. Poudel, S. Liwicki, and R. Cipolla, "Embodied visual navigation with automatic curriculum learning in real environments," in *IEEE Robotics and Automation Letters (RA-L)*, 2021.

[16] B. Shen, D. Xu, Y. Zhu, L. J. Guibas, F. Li, and S. Savarese, "Situational fusion of visual representation for visual navigation," in *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 2881–2890, 2019.

[17] A. Sax, B. Emi, A. R. Zamir, L. J. Guibas, S. Savarese, and J. Malik, "Mid-level visual representations improve generalization and sample efficiency for learning visuomotor policies.," *arXiv preprint arXiv:1812.11971*, 2018.

[18] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. Rehg, B. Boots, and E. Theodorou, "Information Theoretic MPC for Model-Based Reinforcement Learning," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2017.

[19] R. Kusumoto, L. Palmieri, M. Spies, A. Csiszar, and K. O. Arras, "Informed information theoretic model predictive control," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2047–2053, IEEE, 2019.

[20] S. Bansal, V. Tolani, S. Gupta, J. Malik, and C. Tomlin, "Combining optimal control and learning for visual navigation in novel environments," in *Proc. of the Conf. on Robot Learning (CoRL)*.

[21] J. Francis, N. Kitamura, F. Labelle, X. Lu, I. Navarro, and J. Oh, "Core challenges in embodied vision-language planning," *Journal of Artificial Intelligence Research*, vol. 74, pp. 459–515, 2022.

[22] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[24] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations," *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.

[25] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niebner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," in *Proc. of the Int. Conf. on 3D Vision (3DV)*, 2018.

[26] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A platform for embodied ai research," in *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 9338–9346, 2019.

[27] V. Cartillier, Z. Ren, S. Jain, N.and Lee, I. Essa, and D. Batra, "Semantic mapnet: Building allocentric semantic maps and representations from egocentric views," *arXiv preprint arXiv:2010.01191*, 2020.

[28] P. Anderson, A. Chang, D. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecká, J. Malik, R. Mottaghi, M. Savva, and A. Zamir, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.