

Probable Object Location (POLo) Score Estimation for Efficient Object Goal Navigation

Jiaming Wang¹ and Harold Soh^{1,2}

Abstract—In this work, we focus on object search tasks within unexplored environments. We introduce a framework centered around the Probable Object Location (POLo) score. Utilizing a 3D object probability map, the POLo score allows the agent to make data-driven decisions for efficient object search. We further enhance the framework’s practicality by introducing POLoNet, a neural network trained to approximate the computationally-intensive POLo score. Our approach addresses critical limitations of both end-to-end reinforcement learning methods, which suffer from memory decay over long-horizon tasks, and traditional map-based methods that neglect visibility constraints. Our experiments, involving the first phase of the Open-Vocabulary Mobile Manipulation (OVMM) 2023 challenge, demonstrate that an agent equipped with POLoNet significantly outperforms a range of baseline methods, including end-to-end RL techniques and prior map-based strategies. To provide a comprehensive evaluation, we introduce new performance metrics that offer insights into the efficiency and effectiveness of various agents in object goal navigation.

I. INTRODUCTION

Efficiently searching for objects in unexplored environments is a crucial skill for intelligent agents in real-world scenarios. Humans excel at solving this partially observable task by employing two key strategies: 1) leveraging common object associations, such as finding a coffee mug near a coffee machine [1], [2]; 2) employing efficient search behaviors like glancing into unexplored rooms while passing by and thoroughly examining promising areas where the target object is likely situated. This study concentrates on the latter aspect: devising a method to equip autonomous robots with efficient search behaviors.

Current state-of-the-art methods for this task can be broadly classified into two categories: map-based methods and non-map-based methods. Non-map-based methods [3], [4], [5] often rely on end-to-end learning paradigms, leveraging implicit memory structures like RNNs or LSTM networks to handle the partial observability of the environments. While effective in simple scenarios where the target objects are conspicuous and the environment is limited in size, these methods encounter difficulties in complex environments. It has been shown that information stored in the memory cell in baseline LSTMs decays exponentially [6], [7], potentially leading to a decline in spatial awareness for the agent. This results in agents re-exploring areas they have already traversed, ultimately causing a failure to locate the target object within the allocated time constraints.

Conversely, map-based methods tackle the challenge by concurrently constructing a spatial representation of the environment as the robot explores. This constructed map serves as an explicit memory for the agent, providing the advantage of enhanced generality and interpretability. However, it also necessitates the development of efficient search strategies. Existing search methodologies in this area often adopt a two-stage heuristic: initial map exploration followed by object-specific search. However, these methods typically overlook critical considerations such as what is visible from a specific map position [8], [9] and prioritized exploration of promising areas [10], [11], [12], [13], which in turn leads to suboptimal search trajectories.

In this paper, we introduce a map-based framework to address the above limitations (Fig. 1). Central to our framework is the introduction of a new metric called the *Probable Object Location* (POLo) score. The POLo score serves to: (i) quantify the likelihood of an object being located within a 3D environment, thus guiding the agent to focus on high-probability areas while still taking into account visibility constraints; (ii) promote the exploration of as-yet-unmapped regions. However, computing the POLo score can be computationally expensive as it requires consideration of visible areas (akin to rendering).

To address this computational bottleneck, we introduce an approximation technique using a trained neural network that we term POLoNet. This neural network is engineered to estimate the POLo score efficiently, without significantly compromising accuracy. The integration of POLoNet within our framework enables a robot to make real-time decisions in complex environments, thereby mitigating the computational cost associated with such comprehensive scores.

We validate our framework using the first phase of the OVMM 2023 challenge, and our empirical results are promising. An agent equipped with the POLoNet framework significantly outperforms various baseline approaches; it achieves better performance compared to an end-to-end Reinforcement Learning (RL) method and previously proposed map-based strategies. Specifically, our agent demonstrates enhanced performance in terms of success rate and efficiency metrics like Success weighted by Path Length (SPL). This evidence substantiates the robustness and effectiveness of the POLo score and POLoNet. In brief, our main contributions are as follows:

- We propose a map-based framework for efficiently searching for objects in an unknown environment. To the best of our knowledge, this is the first work to incorporate a “next-best-view”-like strategy for ob-

¹Jiaming Wang and Harold Soh are with Smart Systems Institute, National University of Singapore. jamie-w@nus.edu.sg

²Dept of Computer Science, National University of Singapore. harold@comp.nus.edu.sg

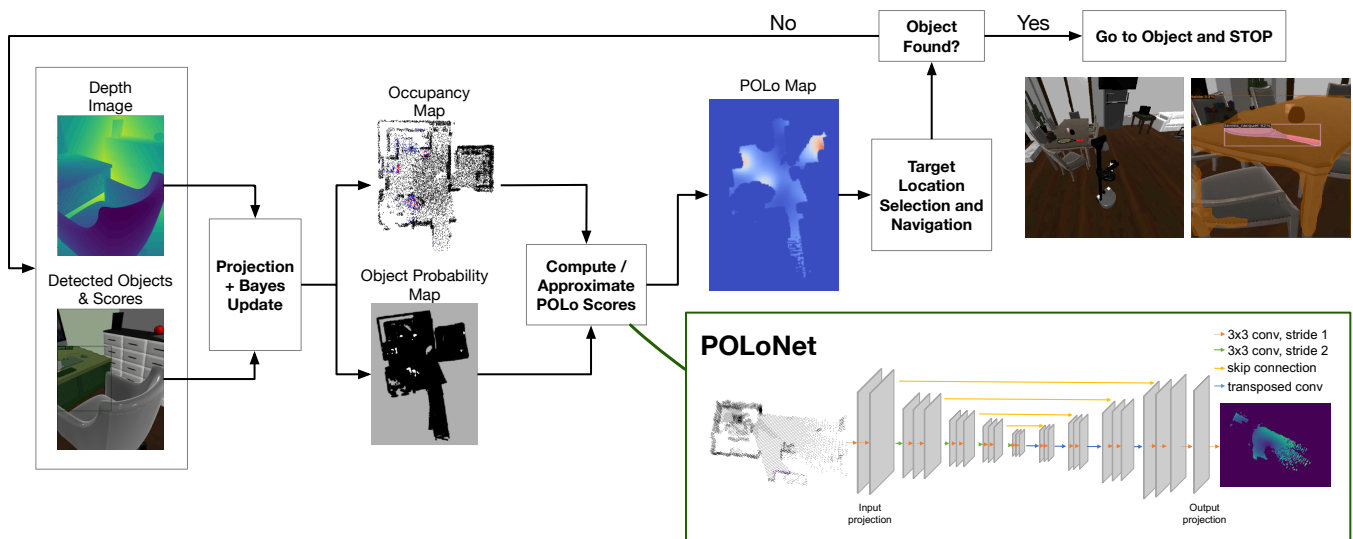


Fig. 1. Overview of our POLo Robot Agent for Object Goal Navigation. Here, the robot is tasked to find a tennis racquet on a table. The robot has a depth camera and uses an object detector (DETI [14] in our case). It maintains an occupancy map and an object probability map, which are used to compute a POLo score for each robot configuration — this score reflects the agent’s belief of an object being located within the viewable areas from a certain configuration. Since computing the POLo score is expensive (it requires computing which voxels are visible), we approximate it using POLoNet, a prediction network that takes as input a tensor with dimensions $15 \times 192 \times 192$, where the height dimension (15) serves as the channel dimension for 2D convolution. This tensor is created through cropping and down-sampling of the current global occupancy map \mathcal{M}^o and the probabilistic map \mathcal{M}^p . POLoNet produces an output tensor of size $2 \times 192 \times 192$, representing a dense POLo score map. The robot selects the configuration with the highest POLo score as the next target location and navigates to the area.

ject goal navigation tasks. Experiments show that our POLoNet framework significantly outperforms other baseline methods.

- We conduct a comprehensive set of experiments and analyses on the proposed method and multiple baseline methods. We introduce new metrics for evaluating object navigation tasks, providing insights into the behavior of alternative methods for object goal navigation.

II. TASK DEFINITION

In the Object Goal Navigation [9], [15] task, the agent’s objective is to locate an object specified by its object class name. At each step, the agent receives egocentric RGB and depth images, in addition to localization (GPS) and compass readings. The action space consists of four discrete actions: MOVE FORWARD, TURN LEFT, TURN RIGHT, and STOP. An episode is deemed successful when the agent confirms object discovery by executing the STOP action within a certain threshold distance d_s from the target object. The episode terminates after a fixed number of timesteps.

To evaluate and compare different methods, we adopt the object goal navigation sub-task from the 2023 Habitat Open Vocabulary Mobile Manipulation (OVMM) challenge [16]. The challenge defines goals using natural language, e.g., “move an apple from the table to the counter”. The OVMM challenge comprises four distinct sub-tasks, where the agent is tasked not only with locating objects as described in text but also with executing subsequent object manipulation tasks, including pick and place actions directed towards specified receptacles.

For our study, we focus on the object goal navigation

phase, representing the initial stage within the OVMM challenge. This phase is difficult; unlike the original Habitat object navigation task, the OVMM environment more closely simulates real-world conditions. The OVMM objects are from an open set and are comparatively smaller, such as cups and cellphones, as opposed to the beds and couches. The scenes in the OVMM environment are also larger, compelling the agent to explore its surroundings more efficiently.

III. RELATED WORKS

Object Goal Search With Implicit Memories. There is a growing body of research dedicated to object goal navigation tasks. One research avenue focuses on training agents directly using end-to-end reinforcement learning or imitation learning [17], [18]. Typically, RNNs or LSTMs serve as implicit memory mechanisms to encode the environment. To enhance agent performance and generalization, Maksymets et al. [3] proposed the use of data augmentation by generating additional training episodes with common household objects, while Ye et al. [19] employed auxiliary tasks to facilitate learning. In contrast, Ramrakhya et al. [4] collected a substantial human demonstration dataset and demonstrated that agents trained using behavior cloning on this dataset can acquire efficient searching behaviors. However, end-to-end methods, in general, suffer from low sampling efficiency [20] and limited generalizability when applied to real-world scenarios [21]. Our experiments reveal that a RL agent tended to revisit already searched areas, potentially due to memory decay issues in its RNN.

Object Goal Search with Explicit Maps. Our work utilizes an explicit map to represent the environment, which enables

sample efficiency and potentially better generalization via a modular paradigm [8], [9], [22], [23]. Prior map-based methods use explicit representations as proxies for robot observations and employ either learned or heuristic policies to select high-level goals based on the constructed map. In contrast, our approach incorporates a 3D probabilistic map and a POLO score, allowing the agent to select object locations in a manner that balances exploration and exploitation.

Exploration with Next Best View. A task that is related to object goal search is autonomous exploration. An effective strategy for this task involves employing a “next-best-view” selection strategy aimed at maximizing the coverage of unexplored areas efficiently. This approach optimizes “information gain” along the exploration trajectory, as demonstrated in [10], [11], [12], [13]. In these studies, information gain, $\text{Gain}(\zeta)$, is defined as

$$\text{Gain}(\zeta) = \text{Visible}(\mathcal{M}, \zeta) e^{-\lambda d_\zeta}, \quad (1)$$

where $\text{Visible}(\mathcal{M}, \zeta)$ represents the cardinality of the set of visible and unmapped voxels from the configuration ζ within the global occupancy map \mathcal{M} . d is the path cost from the current configuration to ζ , and tuning factor λ penalizes high path costs. This approach facilitates exploration and coverage of the environment, but is not directly applicable to object goal navigation since it does not explicitly inspect promising regions where the target object is likely to be situated.

IV. POLO SCORE FOR OBJECT GOAL NAVIGATION

When searching an object in an unknown environment, humans handle uncertainty by considering potential object locations. For instance, when encountering an unexplored room, we briefly investigate it. If we notice anything indicative of the object’s presence, we examine further. Otherwise, we move on to explore other areas. This behavior suggests a representation of the environment that captures *probable object locations* based on previous observations, and a strategy that considers both exploration and exploitation.

With this in mind, we developed an agent equipped with a 3D representation to track probable locations of the object, supplementing the conventional occupancy map used in prior work (Fig. 1). To optimize the search using this map, we introduce a measure known as the POLO score. This score quantifies the likelihood of an object’s presence within visible voxels, thereby enabling the agent to explore areas of high probability without neglecting visibility constraints. However, calculating the POLO score is computationally expensive. To mitigate this issue, we propose an approximation technique using a trained neural network, termed POLONet. Subsequent sections elaborate on each of these components, beginning with the probabilistic map, followed by discussions on the POLO score and POLONet.

A. 3D Target Object Probability Map

As stated above, the robot maintains both a 3D object probability map, represented as \mathcal{M}^p , and an occupancy map, denoted as \mathcal{M}^o . This is in contrast to relying solely on

a 2D/3D binary occupancy map as in prior object navigation [8], [9], [24], and map coverage tasks [10], [11], [12], [13]. This 3D probabilistic map represents regions that are likely to contain the target object. It is represented as a tensor with dimensions $\mathcal{M}^p \in \mathcal{R}^{L \times L \times H}$, where L represents the size of the 3D map, and H denotes the height. Each voxel corresponds to a 5cm^3 physical space in the environment.

The map is updated at each time-step; we generate an ego-centric local map $\mathcal{M}_L \in \mathcal{R}^{N \times N \times H}$ using the current depth image and detection results obtained from an open vocabulary detector [14]. Each entry in the local map corresponds to a detection likelihood $p(v_i|o^T)$, which represents the likelihood of an object of interest in voxel v_i given the observation o at timestep T . The likelihood function is defined as $p(v_i|o^T) = g(s_{\text{goal}}, s_{\text{recep}})$, where s_{goal} and s_{recep} are the detection scores for the goal object and its receptacle, respectively (as specified by the language instructions). In our work, g is a distance-based weighted combination of the detection scores, but alternative functions can be used. Subsequently, we perform a spatial transformation to convert this local map into the global frame and update the global map using Bayes rule:

$$p(v_i|o^T, o^{1:T-1}) = \frac{p(v_i|o^{1:T-1})}{1 - p(v_i|o^{1:T-1})} \times \frac{p(v_i|o^T)}{1 - p(v_i|o^T)} \quad (2) \\ \times \frac{1 - p(v_i)}{p(v_i)} \times (1 - p(v_i|o^T, o^{1:T-1}))$$

where $p(v_i)$ denotes the prior probability. Note that our map assumes that the probability of the an object in a specific voxel is independent of other voxels (once conditioned upon the observation). Although this may not be true, this independence assumption allows us to efficiently maintain this probabilistic map, as each voxel can be updated independently.

In preliminary experiments, we attempted to directly use this 3D map by selecting the voxel with the highest probability as the target goal location. However, this purely exploitation-oriented approach yielded suboptimal results. The limitations appear to stem from two primary factors: (i) noisy detections affecting the accuracy of the probabilistic map, and (ii) an insufficient exploration strategy that failed to investigate other potentially viable areas. To rectify these shortcomings, we adopted a more balanced methodology, which involves calculating the POLO score, described in the next subsection.

B. POLO Score Calculation

Using the global 3D object probability map, we calculate the **Probable Object Location Score**. Conceptually, the POLO score gauges the “utility” of an *area* surrounding a given robot pose within the map. The score comprises two key elements: (i) an exploration component, which encourages investigation of unexplored regions rich in visible voxels, and (ii) an exploitation component, which favors areas comprising voxels with a high probability of containing

the target object. Formally, POLO is defined as,

$$\text{POLO}(\zeta) = \left(\sum_{v_i \in \mathcal{V}_{<\delta}^{\zeta}} f(p(v_i), \zeta) + \beta \sum_{v_i \in \mathcal{V}_{\geq\delta}^{\zeta}} f(p(v_i), \zeta) \right) e^{-\lambda d_{\zeta}} \quad (3)$$

where $\mathcal{V}_{<\delta}^{\zeta} = \{v \in \mathbf{Visible}(\mathcal{M}^o, \zeta) | p(v_i) < \delta\}$ and $\mathcal{V}_{\geq\delta}^{\zeta} = \{v \in \mathbf{Visible}(\mathcal{M}^o, \zeta) | p(v_i) \geq \delta\}$ and $\mathbf{Visible}(\mathcal{M}^o, \zeta)$ represents the set of visible and unmapped voxels from the configuration ζ within the global occupancy map \mathcal{M}^o . The function $f(\cdot, \zeta)$ is used to adjust probabilities based on the voxel’s location relative to the agent’s configuration. Considering that accuracy in object detection decreases with distance [25], we apply a distance-based decay $f(p(v_i), \zeta) = p(v_i)/d(v_i, \zeta)$, where $d(v_i, \zeta)$ represents the distance between voxel v_i and the configuration ζ .

The POLO score incorporates two summands: the first corresponds to the exploration component, while the second represents the exploitation component. The last exponent multiplier in the equation serves to penalize configurations based on their distance, akin to the **Gain** measure. Two parameters, β and δ , modulate the score’s behavior. Specifically, β dictates the degree to which high-probability voxels are prioritized during the search. Conversely, δ represents our confidence level in the voxel-based object probabilities. A higher δ value fosters a more conservative strategy, as fewer voxels will contribute to the exploitation component in the score calculation.

In our implementation, the agent’s next target location was determined by the configuration $\zeta \in \mathcal{R}^2 \in [0, L] \times [0, L]$ —representing its position on the map—that yielded the highest POLO score. Although the view angle θ could potentially be incorporated into the agent’s configuration ζ , our preliminary experiments showed that omitting θ simplifies computation without negatively impacting performance. Therefore, $\mathbf{Visible}(\mathcal{M}^o, \zeta)$ includes all voxels visible from ζ , under the assumption that the agent has a panoramic (360°) field of view.

C. POLONet

Computing the POLO score is similar to 3D voxel rendering, with probabilities associated with each voxel as features. For these calculations, we utilized the PyTorch3D library [26]. However, calculating POLO scores for all conceivable configurations within the map proved to be excessively time-consuming.

To circumvent this bottleneck, we trained a neural network, termed POLONet, to predict POLO scores (Fig. 1). The network takes as input a cropped global voxel map, denoted as $\mathcal{M}^{crop} \in \mathcal{R}^{N' \times N' \times H}$, derived from the global occupancy map \mathcal{M}^o and the probabilistic map \mathcal{M}^p . In this tensor, each voxel has value -1 if unobserved or $p(\mathbf{V}_i)$ otherwise.

Our network architecture design is based on achieving a large receptive field covering the agent’s maximum depth range of 10m. This enables POLO score prediction by considering the surrounding 3D structures. We employ a U-Net-like network structure with skip connections, which

TABLE I
POLONET ERROR SCORES WITH ABLATIONS

Method	Test MAE ↓
POLONet	0.0117 ± 0.0468
w/o SC	0.0258 ± 0.1270
w/o DA	0.0190 ± 0.1257

enables the incorporation of both coarse and fine-grained features. Instead of employing expensive 3D convolution, we use 2D convolutions by treating the height dimension as a channel dimension. The network outputs a dense POLO map $\mathcal{M}^{\text{POLO}} \in \mathcal{R}^{2 \times N' \times N'}$, with the two layers in the output tensor corresponding to the two terms in (3).

As a preliminary evaluation, we trained POLONet on ground truth labels calculated using Eqn. (3). Overall, we generated 19,726 data pairs $((\mathcal{M}^o, \mathcal{M}^p), \mathcal{M}^{\text{POLO}})$, with each label $\mathcal{M}^{\text{POLO}}$ containing POLO scores sparsely rendered at 150 randomly chosen locations due to computational budget limitations. To optimize learning from this constrained dataset, we applied data augmentation techniques such as random translation and rotation during training.

POLONet was tested with 1,464 novel data pairs. We compared POLONet against two variants: (i) one without skip connections (w/o SC), and (ii) one without data augmentation (w/o DA). Results in Table I reveal that POLONet achieves a low test Mean Absolute Error (MAE). Both skip connections and data augmentation contributed to error reduction. Remarkably, the average computation time for POLONet stood at only 0.26 ± 0.06 seconds—a 76-fold speedup compared to the conventional POLO score computation, which required 20 ± 3.23 seconds¹.

V. EXPERIMENTS AND RESULTS

In this section, we report on experiments on object goal navigation using the Habitat simulator [27]. Our primary goal was to validate if the POLO score and its estimation using POLONet led to performance gains. We evaluated POLONet and other baselines using 200 episodes generated from 12 novel scenes in the test dataset. An episode was successful if the target object was found within 1250 simulation steps, and considered unsuccessful otherwise.

A. Experimental Setup

Model Training. In our experimental evaluation, we trained the POLO prediction network (POLONet) using 38 scenes from the training dataset using the techniques described in the previous section.

Baselines and Variants. We benchmarked the performance of POLONet against four baseline agents and two variants of POLONet, each implementing distinct navigation and search strategies:

- **RL:** This agent, trained end-to-end with DD-PPO [20], utilizes current depth images and segmentation masks

¹Computed for a uniform sample of 150 locations within navigable, mapped areas

TABLE II
EVALUATION METRICS ON THE HABITAT SIMULATOR

Method	Success (%) \uparrow	SPL \uparrow	E/D \uparrow	CP/D \uparrow
RL	12.50	0.060	1.91	0.014
FBE	10.99	0.049	0.98	0.010
Max Coverage	12.95	0.051	3.49	0.021
Max Object Prob.	10.94	0.054	3.80	0.040
POLo-2D	13.20	0.050	4.48	0.033
POLo-R ($\beta = 5$)	17.71	0.066	3.17	0.040
POLoNet ($\beta = 1$)	16.23	0.055	3.30	0.037
POLoNet ($\beta = 3$)	17.71	0.083	3.35	0.040
POLoNet ($\beta = 5$)	20.31	0.083	3.52	0.044

of detected target objects to predict the next discrete action.

- **FBE:** Employing a frontier-based exploration strategy [28], this agent maintains and updates a 2D semantic map at each step and advances toward the goal object upon identification.
- **Max Coverage:** The next high-level goal for this agent is selected based solely on the highest Gain score, calculated from the 3D global occupancy map, in a manner similar to prior work [11].
- **Max Object Probability:** This agent also selects the next high-level goal based solely on the highest probability voxel in the 3D object probability map.

and the following variants:

- **POLo-2D:** This variant calculates the next high-level goal by considering 2D occupancy and 2D probabilistic maps, obtained by projecting the original 3D maps onto the XY plane.
- **POLo-R:** This agent computes the POLo score using PyTorch3D, but restricts computations to a uniform sample of 150 locations within mapped and navigable areas.

For RL and FBE agents, we used the implementation and pre-trained weights from [29]. We also experimented POLoNet agents with different β values.

Metrics. We employed the following metrics to assess agent performance:

- **Success Rate:** This metric represents the ratio of successful episodes to the total number of episodes evaluated.
- **SPL (Success weighted by Path Length):** This metric measures the efficiency of the agent in reaching the target object, as outlined in [30].
- **E/D Ratio (Explored Area divided by Total Traveled Distance):** This metric evaluates the agent’s proficiency in exploration and map coverage.
- **CP/D Ratio (Closely Checked Promising Area divided by Total Traveled Distance):** Promising regions are defined as detected receptacles potentially containing the goal object. This metric quantifies the agent’s efficiency in exploitation by assessing the thoroughness in inspecting promising areas.

These latter two metrics furnish additional insights into agent performance, complementing the standard metrics conventionally employed in object goal navigation tasks.

B. Results and Discussion

Comparisons against Baselines. The results from our experiment are summarized in Table II. The proposed POLoNet with $\beta = 5$ agent outperforms all the baselines by a significant margin in terms of success rate. Moreover, it demonstrates higher scores in terms of the SPL and CP/D ratios, indicating more efficient navigation and exploitation of the promising regions. Qualitatively, we observed that the POLoNet guides the agent towards check promising areas

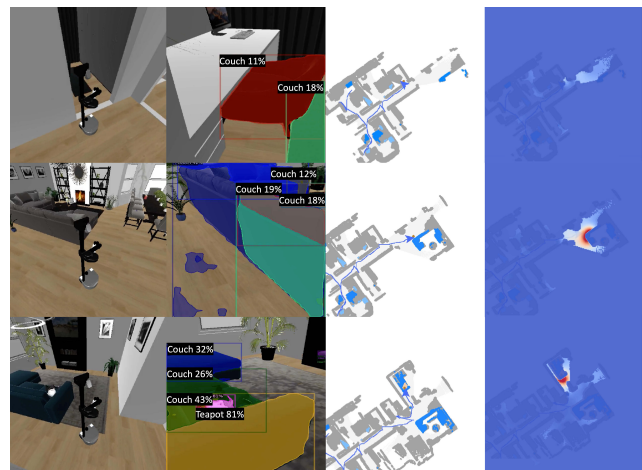


Fig. 2. The exploration trajectory (top to bottom) of finding a teapot on a couch. From left, the first image displays the third person view of the agent, the second image is the agent’s RGB observation with overlaid object detection results. The third image shows the constructed map, along with the agent’s location (blue pointer) and its travelled paths (blue curves). The last image is the predict values from the POLoNet, overlaid with the map. High POLo values are in red. We observe that the agent is guided by the POLo scores and checks multiple rooms by briefly peeking into it (top). When it detects a couch, the POLo score leads the agent to closely check the couch by predicting high values around the couch (middle). The agent continues to explore the map, and successfully finds the teapot (bottom).

and exploring unmapped territory. Fig. 2 visualizes an example trajectory using POLoNet, showing RGB observations, detected objects, projected 2D semantic maps, and predicted POLo maps at different time points in an episode. We can observe that the agent displays efficient search behaviors such as peeking into rooms and closely checking potential receptacles.

The behavior of the RL agent highlights the challenges associated with long-term planning and exploration in object goal navigation. While the RL agent demonstrates efficient low-level control and good initial exploration, its performance drops in longer episodes. The limitations in its implicit memory system result in repetitive, unproductive exploration patterns. This finding is illustrated in Fig. 3, where the exploration rate plateaus after around 800 steps. In contrast, the POLoNet agent maintains a consistent level of efficient exploration throughout the episode, as demonstrated qualitatively in Fig. 4. This suggests that the POLoNet provides a more effective framework for exploration and

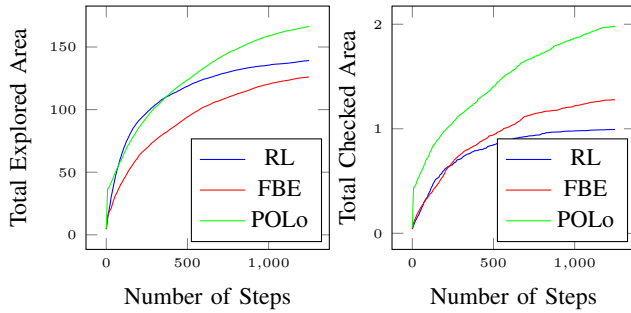


Fig. 3. Left: Accumulated explored (mapped) area at each step, averaged for 200 evaluated episodes. Right: Accumulated closely checked promising area at each step, averaged for 200 evaluated episodes.

exploitation over long time horizons.

The limitations of the Frontier-Based Exploration (FBE) strategy stem from its lack of contextual awareness about what to observe in the environment. While FBE excels at identifying unexplored areas by locating frontiers, it lacks the sophistication to differentiate between areas with differing levels of importance or information content. This often results in suboptimal behavior, for example, choosing inefficient intermediate goals on the frontier. This highlights the importance of using a more structured and information-rich criterion like POLo for guiding agent behavior.

With regards to methods that maximize coverage or maximize object probability, these are strategies that emphasize solely exploration or exploitation, respectively. Neither do particularly well compared to POLoNet, which suggests a middle-ground approach that balances the two criteria is important.

POLoNet Variants. The 2D POLo agent has relatively high E/D and CP/D ratios, indicating the advantages of employing POLo scores even in a 2D context. However, its lower success rate compared to the 3D POLoNet variants highlights the limitations of 2D maps in POLo calculations; a 2D-based approach cannot differentiate between a table and a wall. The tuning of the β parameter affects the CP/D ratio and success rates; a higher β encourages the agent to scrutinize promising areas before venturing into unexplored territories. This is particularly beneficial in expansive environments where transitioning between rooms incurs a higher cost.

Error Analysis. The success rates of all methods in our experiment are notably lower compared to the success rates reported in the Habitat Object Goal Navigation Challenge (HON) [31]. This discrepancy arises from the greater difficulty of the OVMM environment. All the agents encounter failures stemming from faulty perception due to small target objects. Notably, the success rate of the POLoNet agent increases to 56.7% when employing ground truth perception. In addition, we observed that the presence of numerous narrow corridors and doors within the OVMM scenes frequently leads to collisions and failures in the low-level controller employed by modular approaches. We aim to address these issues as part of our future work.

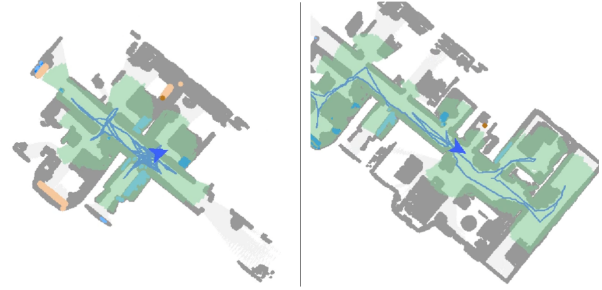


Fig. 4. Exploration trajectories represented by blue curves for the RL agent (left) and the POLoNet agent (right), for the same episode. The RL agent explores a smaller area within an equivalent number of steps due to its tendency to repeatedly visit the same room.

VI. CONCLUSIONS

In this study, we introduce a framework enabling autonomous agents to efficiently search for objects in unknown environments. Our proposed method maintains a continuously updated 3D probabilistic map and quantifies POLo Score based on this map to guide the search process. The POLo score provides the agent with a data-driven basis for decision-making, allowing it to focus on high-probability zones without overlooking visibility constraints. Recognizing the computational burden of calculating the POLo score, we introduced POLoNet, a neural network designed to approximate this score efficiently.

Our empirical evaluation, undertaken within the context of the OVMM 2023 challenge, demonstrates that an agent equipped with POLoNet outperforms a variety of baseline methodologies. Our work also contributes to the research community by introducing new performance metrics that allow for a more nuanced understanding of an agent’s efficiency and effectiveness in object goal navigation tasks. Moving forward, we are working to transfer these successes to real-world robotic systems, accounting for sensor noise, occlusion, and other real-world challenges.

ACKNOWLEDGMENT

This research is supported by the National Research Foundation, Singapore under its Medium Sized Center for Advanced Robotics Technology Innovation.

REFERENCES

- [1] G. Sarch, Z. Fang, A. W. Harley, P. Schydlo, M. J. Tarr, S. Gupta, and K. Fragkiadaki, “TIDEE: Tidying Up Novel Rooms Using Visuo-Semantic Commonsense Priors,” in *Computer Vision – ECCV 2022 - 17th European Conference, Proceedings*, ser. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Germany: Springer, 2022, pp. 480–496.
- [2] M. Narasimhan, E. Wijmans, X. Chen, T. Darrell, D. Batra, D. Parikh, and A. Singh, “Seeing the un-scene: Learning amodal semantic maps for room navigation,” in *European Conference on Computer Vision*, 2020.

- [3] O. Maksymets, V. Cartillier, A. Gokaslan, E. Wijmans, W. Galuba, S. Lee, and D. Batra, "THDA: Treasure Hunt Data Augmentation for Semantic Navigation," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 15 354–15 363.
- [4] R. Ramrakhya, E. Undersander, D. Batra, and A. Das, "Habitat-Web: Learning Embodied Object-Search Strategies from Human Demonstrations at Scale," in *CVPR*, 2022.
- [5] R. Ramrakhya, D. Batra, E. Wijmans, and A. Das, "PIRLNav: Pre-training with Imitation and RL Finetuning for ObjectNav," in *CVPR*, 2023.
- [6] S. Mahto, V. A. Vo, J. S. Turek, and A. G. Huth, "Multi-timescale representation learning in lstm language models," *arXiv preprint arXiv:2009.12727*, 2020.
- [7] C. Tallec and Y. Ollivier, "Can recurrent neural networks warp time?" in *International Conference on Learning Representation 2018*, 2018.
- [8] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman, "PONI: Potential Functions for ObjectGoal Navigation with Interaction-free Learning," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, Jun. 2022, pp. 18 868–18 878.
- [9] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, "Object Goal Navigation using Goal-Oriented Semantic Exploration," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 4247–4258.
- [10] L. Heng, A. Gotovos, A. Krause, and M. Pollefeys, "Efficient visual exploration and coverage with a micro aerial vehicle in unknown environments," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 1071–1078, iSSN: 1050-4729.
- [11] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding Horizon "Next-Best-View" Planner for 3D Exploration," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1462–1468.
- [12] M. Selin, M. Tiger, D. Duberg, F. Heintz, and P. Jensfelt, "Efficient Autonomous Exploration Planning of Large-Scale 3-D Environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1699–1706, 2019.
- [13] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An Efficient Sampling-Based Method for Online Informative Path Planning in Unknown Environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1500–1507, 2020.
- [14] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting Twenty-thousand Classes using Image-level Supervision," Jul. 2022, arXiv:2201.02605 [cs].
- [15] S. Wani, S. Patel, U. Jain, A. X. Chang, and M. Savva, "Multion: Benchmarking semantic map memory using multi-object navigation," in *NeurIPS*, 2020.
- [16] S. Yenamandra, A. Ramachandran, M. Khanna, K. Yadav, D. S. Chaplot, G. Chhablani, A. Clegg, T. Gervet, V. Jain, R. Partsey, R. Ramrakhya, A. Szot, T.-Y. Yang, A. Edsinger, C. Kemp, B. Shah, Z. Kira, D. Batra, R. Mottaghi, Y. Bisk, and C. Paxton, "The home-robot open vocab mobile manipulation challenge," in *Thirty-seventh Conference on Neural Information Processing Systems: Competition Track*, 2023.
- [17] P. Mirowski, M. Grimes, M. Malinowski, K. M. Hermann, K. Anderson, D. Teplyashin, K. Simonyan, A. Zisserman, R. Hadsell *et al.*, "Learning to navigate in cities without a map," *Advances in neural information processing systems*, vol. 31, 2018.
- [18] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 4693–4700.
- [19] J. Ye, D. Batra, A. Das, and E. Wijmans, "Auxiliary tasks and exploration enable objectgoal navigation," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16 097–16 106, 2021.
- [20] E. Wijmans, A. Kadian, A. S. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames," in *International Conference on Learning Representations*, 2019.
- [21] T. Gervet, S. Chintala, D. Batra, J. Malik, and D. S. Chaplot, "Navigating to objects in the real world," *Science Robotics*, vol. 8, 2022.
- [22] S. Gupta, V. Tolani, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," *International Journal of Computer Vision*, vol. 128, pp. 1311 – 1330, 2017.
- [23] G. Georgakis, B. Bucher, K. Schmeckpeper, S. Singh, and K. Daniilidis, "Learning to map for active semantic goal navigation," in *International Conference on Learning Representations*, 2021.
- [24] T. Chen, S. Gupta, and A. Gupta, "Learning exploration policies for navigation," in *International Conference on Learning Representations*, 2018.
- [25] Y. Hao, H. Pei, Y. Lyu, Z. Yuan, J.-R. Rizzo, Y. Wang, and Y. Fang, "Understanding the impact of image quality and distance of objects to object detection performance," 2022.
- [26] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari, "Accelerating 3d deep learning with pytorch3d," *arXiv:2007.08501*, 2020.
- [27] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets, A. Gokaslan, V. Vondruš, S. Dharur, F. Meier, W. Galuba, A. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, and D. Batra, "Habitat 2.0: Training Home Assistants to Rearrange their Habitat," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 251–266.
- [28] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97: Towards New Computational Principles for Robotics and Automation*. IEEE, 1997, pp. 146–151.
- [29] S. Yenamandra, A. Ramachandran, K. Yadav, A. Wang, M. Khanna, T. Gervet, T.-Y. Yang, V. Jain, A. W. Clegg, J. Turner *et al.*, "Homerobot: Open-vocabulary mobile manipulation," *arXiv preprint arXiv:2306.11565*, 2023.
- [30] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.
- [31] K. Yadav, S. K. Ramakrishnan, J. Turner, A. Gokaslan, O. Maksymets, R. Jain, R. Ramrakhya, A. X. Chang, A. Clegg, M. Savva, E. Undersander, D. S. Chaplot, and D. Batra, "Habitat challenge 2022," <https://aihabitat.org/challenge/2022/>, 2022.