2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Learning Semantic Priorities for Autonomous Target Search

Max Lodel, Nils Wilde, Robert Babuška, Javier Alonso-Mora

***Abstract*— The use of semantic features can improve the efficiency of target search in unknown environments for robotic search and rescue missions. Current target search methods rely on training with large datasets of similar domains, which limits the adaptability to diverse environments. However, human experts possess high-level knowledge about semantic relationships necessary to effectively guide a robot during target search missions in diverse and previously unseen environments. In this paper, we propose a target search method that leverages expert input to train a model of semantic priorities. By employing the learned priorities in a frontier exploration planner using combinatorial optimization, our approach achieves efficient target search driven by semantic features while ensuring robustness and complete coverage. The proposed semantic priority model is trained with several synthetic datasets of simulated expert guidance for target search. Simulation tests in previously unseen environments show that our method consistently achieves faster target recovery than a coverage-driven exploration planner.**

I. Introduction

Autonomous robots that can explore unknown environments efficiently by searching for objects of interest (OOI) are promising tools in applications such as search and rescue, inspection, and environmental monitoring. Efficient search typically relies on reasoning about semantic information in the scene and consequently determining *where to search* next. For example, search and rescue in an industrial site likely focuses on zones frequently used by workers, such as offices and storage rooms, that can be identified by characteristic objects like desks or shelfs.

By leveraging semantic priors of typical object arrangement, recent works [1]–[8] have demonstrated this semantic exploration paradigm and achieved effective autonomous search behavior. However, these methods either train on large domain-specific datasets [9], [10] or use foundation models trained on internet-scale datasets, leading to common-sense reasoning capabilities [5]–[7].

However, domain-specific training data may not always be available for highly unpredictable and specific environments. Moreover, foundation models require extensive computational resources that are infeasible for onboard deployment. On the contrary, pure coverage exploration methods [11]–[13] that effectively search everywhere can be deployed independently of domain priors but can take a long time to find OOIs.

This work was supported by the National Police of the Netherlands. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

The authors are with the Department of Cognitive Robotics (CoR), Delft University of Technology, The Netherlands, {m.lodel; n.wilde; r.babuska; j.alonsomora}@tudelft.nl. R. Babuška is also with CIIRC, Czech Technical University in Prague, Czech Republic.

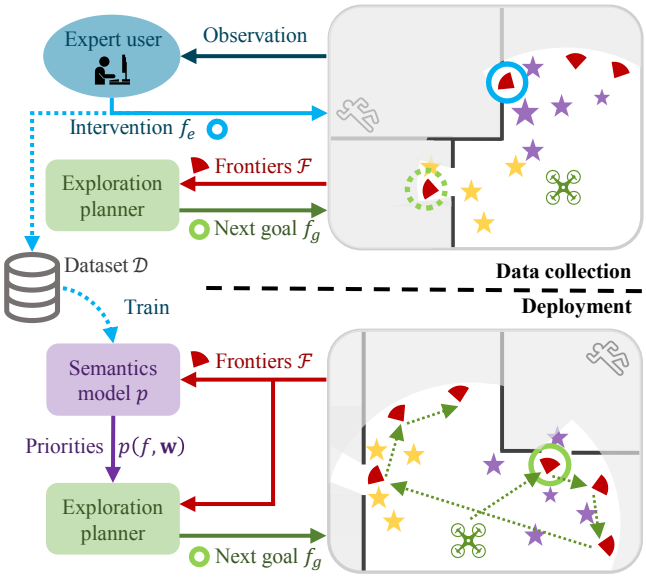

Fig. 1: Conceptual overview of the proposed framework. During data collection, an expert generates interventions into the planner's goal output, prioritizing certain semantically relevant objects (depicted by stars). These are used to train a semantic model, which outputs priorities for each exploration frontier that, in turn, guide the exploration planner. The exploration planner outputs the next frontier viewpoint to navigate to.

Specialized human operators, such as first responders in search and rescue, often have high-level knowledge about promising search locations based on observed semantic features, such as relevant objects. However, increasing autonomy in exploration can be prefereable over teleoperation, as it reduces the operator's workload and is less reliant on robust communication. Hence, we aim to leverage expert inputs to learn semantic priors for autonomous target search. Independent of semantic information, coverage-driven exploration methods [11]–[13] can guarantee target discovery. Therefore, a reliable semantic search approach must ensure that efficient exploration of the entire environment continues independently of semantic features and their learned priorities.

In this paper, we present a hierarchical exploration framework (Fig. 1), that can learn semantics target search strategies from expert inputs. Our paper makes the following contributions:

- We introduce a framework for learning a semantic priority function that models the knowledge driving expert interventions, instead of imitating the expert.
- We present a novel exploration planner leveraging these priorities to prioritize promising frontiers.

In simulation experiments, we show that our framework achieves more efficient target search than coverage exploration after learning from only a small set of expert interventions.

Moreover, our approach exhibits robust target search performance when learning from different simulated expert behaviors.

II. Related Works

In this section, we discuss existing approaches and how they relate to our work, focussing on semantic target search, learning human objective functions and coverage exploration.

1) Semantically-informed Target Search: Semantically informed target search exploits environmental semantic features to accelerate target localization. Several works address object search in unknown environments by learning semantic object relations from large-scale datasets [9], [10]. Reinforcement learning (RL) approaches [1], [2] train target search policies directly in simulation, whereas other methods [3], [4] predict the cost-to-go of different positions, demonstrating better data efficiency than RL. Conversely, zero-shot object search [5]–[7] show that foundation models trained on internet-scale data can be used to predict likely object locations from semantic context in common indoor environments. The authors of [8] distill semantic knowledge from a large language model (LLM) into a smaller model for online inference of target probabilities. In our paper, we learn a model of semantic priorities, similar to the prediction approaches [3], [4]. We learn semantic knowledge from expert inputs unlike prior work using environment data [1]–[4], comparable to distilling LLM common sense reason in [8].

2) Learning Human Objectives: Learning a priority model from human feedback involves learning the human's objective function. In most works, this is formalized as learning a reward function [14]–[16] or action-value function [17], [18]. *Offline* feedback methods [14], [16], [17] query the human for choice of different precomputed system behaviors [14], [16] or with states requiring a goal demonstration [17]. However, generating such queries is challenging in uncertain long-horizon tasks like exploration. With *online* feedback [15], [18], the human chooses when to provide inputs as he interacts with an agent executing some baseline behavior. Such online inputs can be binary feedback [15] or interventions with low-level demonstrations [18]. Our method considers online feedback in the form of expert interventions, similar to [18], demonstrating the preferred exploration frontiers. We propose to learn an exploration priority model of different frontiers, similar to learning a value function over planning goals [17]. Moreover, we employ a stochastic model of expert actions, as in [14].

3) Coverage-driven exploration: Coverage-driven exploration methods maximize the expected area coverage in order to build an occupancy map without considering semantic features. Recently proposed methods employ combinatorial planning to visit all exploration frontiers [11]–[13], [19], or navigation policies trained to maximize future coverage rewards using RL [20], [21]. Combinatorial planners repeatedly compute tours over all frontiers, allowing reasoning over long horizons and efficient navigation across frontiers. These approaches have proven to work robustly in challenging real-world experiments [11]–[13], [19]. We build on this concept and employ a combinatorial planner over frontiers, but consider semantic features for target search in the planner. To this end, we propose a planner formulation that, similar to [13], can schedule frontiers based on a priority measure but prioritizes based on both semantics and coverage.

III. Problem Formulation

We consider the usecase where an autonomous robot searches for a target object in an unknown environment $\mathcal{W} \subset \mathbb{R}^2$ with obstacle-free space $\mathcal{W}_f \subset \mathcal{W}$. The robot's position at time t is denoted by $\mathbf{x}_t \in \mathcal{W}_f$, and it starts exploring from an initial position $\mathbf{x}_0$. The robot moves incrementally with actions $\mathbf{a} \in \mathbb{R}^2$ bounded by $|\mathbf{a}| < \delta_{\text{max}}$, where δ_{max} is the maximum distance per time step. An action $\mathbf{a}$ can only be applied if the new position is in free space, i.e., $\mathbf{x}_t + \mathbf{a} \in \mathcal{W}_f$. A more complex dynamic model can be considered, for example, by tracking the reference $\mathbf{a}$ with a model predictive controller, as in [21].

Occupancy Map From range observations with sensing range r until time t, the robot builds an occupancy map $\mathbf{M}_t$ of the environment. The occupancy map is represented as a grid, where cells correspond to evenly spaced positions $\mathbf{x} \in \mathcal{M},\ \mathcal{M} \subset \mathcal{W}$, and are in one of three discrete states: unexplored (0), free (1), or occupied by obstacles (-1), i.e., $\mathbf{M} \in \{-1, 0, 1\}^{m \times m}$ with grid size m.

Semantic features: The robot observes objects of different semantic classes $\mathcal{S}$ when exploring the environment. An object is denoted as $o = (o^p, o^s)$, defined by its position $o^p \in \mathcal{W}_f$ and its semantic class $o^s \in \mathcal{S}$. We assume that the robot's sensors can detect objects around the robot within radius r that are not occluded by obstacles. The set $\mathcal{O}_t$ denotes the objects observed up to time t. We further assume that semantic relationships between objects of different classes exist, i.e., the presence of certain objects can indicate an increased or reduced likelihood of other objects being present close by.

Expert input: We further assume the availability of an expert with knowledge of semantic relationships relevant to the search task. Leveraging this knowledge, the expert can infer likely target locations from the observed objects $\mathcal{O}_t$, and guide the robot to the target with waypoint inputs $\mathbf{h} \in \mathcal{W}_f$ that follow some expert policy μ, i.e., $\mathbf{h}_t \sim \mu(\mathbf{h}_t | \mathcal{O}_t)$. From this expert interaction, a dataset $\mathcal{D}$ of expert inputs $\mathbf{h}_t$ with associated observations $\mathbf{M}_t, \mathcal{O}_t$ is recorded.

Problem: given a dataset of human-guided target search trajectories $\mathcal{D}$, the problem is to find a navigation policy $\pi_{\mathcal{D}}$ controlling the robot with $\mathbf{a}_t = \pi_{\mathcal{D}}(\mathbf{M}_t, \mathcal{O}_t)$, that minimizes the distance traveled until discovering a target object o_g, using the map and object memory as current knowledge about the environment. With H as the final time step, the problem is formulated as

$$\begin{aligned} \pi_{\mathcal{D}} = \arg\min_{\pi} & \sum_{t=0}^{H-1} \|\mathbf{x}_{t+1} - \mathbf{x}_t\| \\ \text{s.t.} \quad & \|o_g^p - \mathbf{x}_H\| \leq r \\ & \mathbf{x}_t = \mathbf{x}_{t-1} + \pi(\mathbf{M}_t, \mathcal{O}_t),\ \forall t \in \{1, \dots, H\}, \end{aligned} \tag{1}$$

where the first constraint indicates target discovery at time H.

IV. METHOD

Exploring an unknown environment to search for a target object requires continually solving two subproblems: Semantic scene understanding, or *where is it promising to explore*, and planning, or *where to go next*, given a set of regions to explore. We choose to solve these two problems in a hierarchical framework depicted in Fig. 1 to obtain a data-efficient approach robust to unseen scenarios.

Both subproblems are solved using the concept of frontier exploration [22], which we formalize for our method in Section IV-A. The first problem of semantic scene understanding is formalized as evaluating different frontiers with a semantic priority function. Specifically, we present an approach to learning such a semantic priority function from expert interventions in Section IV-B. To solve the second problem of efficient navigation, we devise a combinatorial target search planner leveraging the learned semantic priority function. Specifically, the planner determines a visitation order such that semantically promising frontiers with increased probability of target discovery are prioritized, thus approximately solving Problem (1).

A. Frontier Exploration

In this section, we describe how our approach formalizes the concept of frontier exploration, drawing inspiration from recent works [12], [13]. Frontiers are the boundaries between explored and unexplored space in $\mathbf{M}_t$ and are used to derive a discrete set of candidate positions for observing unexplored space, called *frontier viewpoints*, that enable efficient exploration planning. To obtain such frontier viewpoints $f \in \mathcal{F}_t,\ \mathcal{F}_t \subset \mathcal{W}_f$ and efficient paths between them, a topological graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ is gradually constructed in the free space of $\mathbf{M}_t$. At every timestep, the graph is expanded using a sampling-based method from [13], ensuring sparsity. We consider every node $v_i \in \mathcal{V}_t$ as a potential frontier viewpoint if sufficient unexplored area is visible from v_i. To this end, we define a *coverage gain* function $\mathcal{I}(v_i)\colon \mathcal{V} \mapsto \mathbb{R}$ that denotes the gain in map coverage when observing frontiers from v_i. Specifically, the coverage gain approximates the expected gain in the covered area by casting a fixed number of equally spaced rays from v_i and averaging the number of visible unexplored cells on each ray. The set of frontier viewpoint nodes $\mathcal{F}_t$ are those with $\mathcal{I}(v_i) > \mathcal{I}_{\text{thres}}$, referred to as *frontiers* $f \in \mathcal{F}_t$. We further assume that $\mathbf{M}_t$ is clustered into regions, e.g., rooms in a building, using a method such as [23], and each frontier is associated with a region.

B. Modeling Expert Frontier Choices

This section formulates a model of expert behavior that will be used to train the semantic priority model. When collecting data, the expert can *intervene* in the robot's exploration behavior at any time t by determining the next *waypoint* that the robot will navigate to.

1) Semantic Priority Function: The expert considers each available frontier $f \in \mathcal{F}_t$ as a potential intervention waypoint, and evaluates how likely exploring a frontier f leads to the target object, based on nearby objects and the expert's semantic knowledge. This evaluation is formalized as a semantic priority function $p(f, \mathbf{w})$. We model this function with a weighted sum where $\mathbf{w} \in [0,1]^n$ are weights on n different features, such as semantic classes. These features form a semantic feature vector $\boldsymbol{\phi}(f)$ for each frontier f. Thus, the priority function can be written as,

$$p(f, \mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi}(f) \tag{2}$$

which is common in preference learning [16], [24] to allow learning from a small number of expert interventions. The weight vector $\mathbf{w}$ used by the expert is unknown and will be estimated from expert inputs.

2) Semantic Feature Vector: The feature vector $\boldsymbol{\phi}(f)$ consists of two parts: Semantic features $\boldsymbol{\phi}_s$ and an auxiliary region novelty feature ϕ_n, i.e., $\boldsymbol{\phi}(f) = [\boldsymbol{\phi}_s(f),\ \phi_n(f)]^T$. Semantic features $\boldsymbol{\phi}_s$ describe the occurrence of different semantic classes around the frontier node f. Each semantic feature needs to capture the presence of the semantic class in the vicinity and in the region of the frontier. Both effects are part of the semantic feature vector: A binary *local* semantic vector $\boldsymbol{\phi}_{s,l}(f) \in \{0,1\}^{|\mathcal{S}|}$ indicating if a class is visible within a small radius around f, and a binary *region* semantic vector $\boldsymbol{\phi}_{s,r}(f) \in \{0,1\}^{|\mathcal{S}|}$ indicating if a class is present in the same region as f. We combine both as $\boldsymbol{\phi}_s(f) = \lambda\boldsymbol{\phi}_{s,r}(f) + (1-\lambda)\boldsymbol{\phi}_{s,l}(f)$ with λ as hyperparameter. Aggregating semantic features in regions allows the search process to exploit the fact that objects are not distributed randomly, but are clustered within functional areas that serve as strong indicators for related objects. The region novelty feature ϕ_n captures the expert's interest in observing semantic information in unexplored regions, and remains 1 unless a small number of objects are observed in the region of frontier f.

3) Expert Intervention Model: When providing online waypoint interventions, the expert's capability to quickly plan over multiple frontiers is limited. Hence, we model the expert behavior with a greedy algorithm for choosing the next frontier. This greedy choice is modeled by a utility function $u(f)$ assigned to each frontier $f \in \mathcal{F}_t$. The expert is also interested in coverage exploration to guarantee search success without relying only on semantic priorities. Furthermore, the expert aims at minimizing the traveled distance until discovering the target object (Problem (1)), which is modeled by discounting the semantic priority by the traveling costs to the frontier. We propose a greedy choice model, maximizing a utility $u(f, \mathbf{w})$, that combines the semantic priority $p(f, w)$ with the coverage gain $\mathcal{I}(f)$ and the distance to the frontier, given by

$$u(f, \mathbf{w}, w_{\mathcal{I}}) = \delta(f)\big(p(f, \mathbf{w}) + w_{\mathcal{I}}\mathcal{I}(f)\big). \tag{3}$$

Here $\delta(f)$ is the distance-based discounting function, defined as $\delta(f) = 1 - \big({}^{d_t(f)}/_{\max_{f' \in \mathcal{F}_t} d_t(f')}\big) + \epsilon$, with $d_t(f)$ expressing the traveling distance from the current position $\mathbf{x}_t$ to f through $\mathcal{G}_t$ and ϵ defining the minimum discounting factor. The utility model in Eq. (3) adds a coverage term weighted by the learnable parameter $w_{\mathcal{I}}$ to the semantic priority p and discounts this extended priority by a factor $\delta(f)$ decreasing with distance to the frontier. Normalizing distances in $\delta(f)$ ensures consistent utility values across different frontier sets $\mathcal{F}$. Finally, the utility function can be written as a linear model

$u(f, \mathbf{w}, w_{\mathcal{I}}) = u(f, \widetilde{\mathbf{w}}) = \widetilde{\mathbf{w}}^T \widetilde{\boldsymbol{\phi}}(f)$ with augmented weights $\widetilde{\mathbf{w}} = [\mathbf{w}, w_{\mathcal{I}}]^T$ and features $\widetilde{\boldsymbol{\phi}}(f) = \delta(f)\big[\boldsymbol{\phi}(f),\ \mathcal{I}(f)\big]^T$.

4) Pairwise Choice Model: Next, we derive a probabilistic model of the expert's frontier choice to learn the expert weights from noisy expert intervention data. We model the expert preference for a frontier $f_e \in \mathcal{F}_t$ as pairwise choices between f_e and all other available frontiers. Hence, the expert prefers frontier f_e if its utility is higher than of all other available frontiers, i.e., if $u(f_e, \widetilde{\mathbf{w}}) \geq u(f, \widetilde{\mathbf{w}}),\ \forall f \in \mathcal{F}_t \setminus \{f_e\}$. The Bradley-Terry model [14], [25] defines the probability of choosing f_i over f_j, denoted by $\mathbb{P}(f_i \succ f_j)$, as a logistic sigmoid function σ of their utility difference, i.e., $\mathbb{P}(f_i \succ f_j) = \sigma(\beta(u(f_i) - u(f_j)))$. Here, β is the rationality parameter modeling uncertainty in the expert's decision-making process. However, this model assumes that probabilities converge to 0 or 1 for large utility differences. We choose to modify this model to account for a residual error probability independent of the utility difference and β, considering cases where the utility model cannot capture potentially complex expert reasoning. Inspired by [16], we define $\rho \in [0, 0.5]$ as a lower bound on the probability of wrong choice independent of the utilities, used to formulate a scaled and shifted sigmoid function σ_ρ:

$$\sigma_\rho(x) = (1 - 2\rho)\sigma(x) + \rho. \tag{4}$$

Then, the probability that the expert chooses f_e over any $f \in \mathcal{F}_t \setminus \{f_e\}$, given weights $\widetilde{\mathbf{w}}$, is modeled as

$$\mathbb{P}(f_e \succ f | \widetilde{\mathbf{w}}) = \sigma_\rho\big(\beta\ \widetilde{\mathbf{w}}^T(\widetilde{\boldsymbol{\phi}}(f_e) - \widetilde{\boldsymbol{\phi}}(f))\big). \tag{5}$$

Here, β and ρ are tunable hyperparameters. This proposed model captures noisy expert waypoint interventions based on the semantic priority function $p(f, \mathbf{w})$.

5) Learning Expert Weights: The final step of the expert model is learning the expert weights from recorded intervention data. Given a set of N choices $\mathcal{C} = \{(f_e^1, f^1), \ldots, (f_e^N, f^N)\}$ from the expert and assuming a uniform prior, we obtain the maximum likelihood estimate of the expert weights given the expert choices using gradient-based optimization, solving

$$\widetilde{\mathbf{w}}_{mle} = \arg\min_{\widetilde{\mathbf{w}}} \sum_{(f_e, f) \in \mathcal{C}} \big[-\log \mathbb{P}(f_e > f | \widetilde{\mathbf{w}})\big], \tag{6}$$

C. Frontier Planning for Priority-Aware Exploration

In this section, we introduce a global planning method for target search given a semantic priority model (Section IV-B).

1) Target Search as Combinatorial Optimization: We extend coverage-maximizing exploration methods that leverage combinatorial planning over frontier viewpoints [11]–[13], [19], by incorporating semantic priorities. The combinatorial planner generates a visitation order, or *tour*, through all known frontier viewpoints. For effective target search, promising frontiers should be scheduled earlier in the tour, such that the distance to the target object is minimized (Eq. (1)). Consequently, we need to minimize the total distance traveled to frontiers with high semantic priority values $p(f, \mathbf{w})$, which are expected to be close to the target. We frame target search as a variant of the Minimum Latency Problem (MLP) [26], denoted as weighted MLP (WMLP), where the planned visitation latencies of the frontiers are weighted using the learned semantic priority model $p(f, \mathbf{w})$.

2) Planner Formulation: We formulate the planning problem over a subset of nodes in the topological graph $\mathcal{G}_t$ composed of the the frontier nodes $\mathcal{F}_t$ and the robot's current node $v_t \in \mathcal{V}_t$, denoted as $\mathcal{F}_t' = \mathcal{F}_t \cup \{v_t\}$. A distance matrix D contains the lengths of the shortest paths through $\mathcal{G}_t$ between all pairs of nodes in $\mathcal{F}_t'$. The tour T is a sequence of all nodes in $\mathcal{F}_t'$ describing the planned visitation order, always starting with the robot node v_t. We denote that frontier node f_i is scheduled at position j in the tour as $T(j) = f_i$ for $j > 0$, while $T(0) = v_t$. Let $P(f)$ be a priority function that assigns each node in $\mathcal{F}_t'$ a priority weight, and $m = |\mathcal{F}_t'|$, then the WMLP objective is

$$\min_T \sum_{i=1}^{m-1} P(T(i)) \sum_{j=1}^{i} D(T(j-1), T(j)). \tag{7}$$

Assuming a unit velocity, this problem minimizes a priority-weighted sum of the visitation latencies of each frontier, favoring earlier visits to high-priority frontiers. The priority function $P(f)$ leverages the learned semantic priorities $p(f, \mathbf{w}_{mle})$ to prioritize regions that likely lead to the target. Combining semantic priorities with expected coverage gain ensures robust exploration when the semantic priorities are ambiguous or incorrect, e.g., when encountering unseen states. Instead of the weighted sum model used in Eq. (3), we propose a heuristic priority function $P(f)$ that always pursues coverage but is biased to semantically important frontiers, which we found more robust for the WMLP planner. Let $p'(f) = {}^{p(f)}/_{p_{max,t}}$ be the normalized semantic priority of frontier f with $p_{max} = \max_{f \in \mathcal{F}_t} p(f)$, then $P(f)$ is given by

$$P(f) = (p'(f, \mathbf{w}_{mle}) + \alpha) \cdot \mathcal{I}(f). \tag{8}$$

Here, $\alpha \in [0, 1]$ is a hyperparameter controlling the trade-off between semantic priority and coverage gain. Note that while we learn the weight vector $\widetilde{\mathbf{w}}_{mle} = [\mathbf{w}_{mle}, w_{\mathcal{I},mle}]^T$, we only use $\mathbf{w}_{mle}$ for inferring frontier priorities, and discard the learned weight $w_{\mathcal{I},mle}$ of the coverage gain. This allows for tuning the balance between target search and coverage to reflect confidence in the learned semantic priority. The normalization of $p(f)$ addresses states where an important frontier only has a single non-zero feature or low feature activations in $\boldsymbol{\phi}(f)$, which can lead to a low-valued semantic priority $p(f)$. By normalizing $p(f)$ by the maximum value in the current state, the combination of semantic priorities and coverage gains proposed in Eq. (8), with a fixed α across different scenarios, becomes more robust.

3) Plan Execution and Control: We now explain how the exploration planner navigates the robot through the environment, which is summarized in Algorithm 1. At every time step, the perception module updates the topological graph, the frontier set, and the robot's position. The tour is replanned whenever the current frontier set $\mathcal{F}_t$ or their coverage gains change (Line 6). In that case, the priorities

Algorithm 1: Prioritized exploration planning

Input : Semantic priority model weights $\mathbf{w}_{mle}$

Init $\mathcal{G}_t \leftarrow \emptyset$, $\mathcal{F}_t \leftarrow \emptyset$, and unexplored map $\mathbf{M}$
foreach *time step t from 1 until* t_{end} **do**
  $\mathbf{M}_t,\ \mathcal{G}_t,\ \mathcal{F}_t,\ v_t \leftarrow$ PERCEPTIONUPDATE()
  if *Target found* ***or*** $\mathcal{F}_t = \emptyset$ **then**
    break
  if $\mathcal{F}_t \neq \mathcal{F}_{t-1}$ ***or*** $\mathcal{I}(f)$ *changed for any* $f \in \mathcal{F}_t$ **then**
    $\mathbf{P} \leftarrow$ FRONTIERPRIORITIES$(\mathcal{F}_t, \mathbf{w}_{mle})$
      ▷ *Computes vector with Eq.* (8) $\forall f \in \mathcal{F}_t$
    $T \leftarrow$ LNSSOLVER $(\mathcal{G}_t,\ \mathcal{F}_t,\ v_t,\ \mathbf{P})$
    $f_g \leftarrow T(1)$ ▷ *Set goal node to next in tour*
  else
    if $v_t = f_g$ **then**
      $f_g \leftarrow$ next frontier in T
  $\mathcal{P} \leftarrow$ SHORTESTPATH$(\mathcal{G}_t, v_t, f_g)$
  Move to next vertex in $\mathcal{P}$

of all current frontiers are updated, and then a new tour T is found by minimizing Eq. (7) using a large neighborhood search (LNS) algorithm [27]. In each iteration, our custom LNS algorithm uses random destruction of up to 30% of the tour and reconstructs it using the cheapest insertion heuristic [28] follwed by a 2-opt swapping search [29]. Given a new tour, the next frontier in the tour is chosen as the subgoal $f_g = T(1)$. If the tour is not recomputed and the subgoal f_g has been reached (Line 11), the next node in T is set as the goal. Otherwise, f_g stays the same. The shortest path to f_g is planned using A* [30] through $\mathcal{G}_t$, and the robot moves to the first node in the path $v_{p,1} \in \mathcal{V}_t$, applying $\mathbf{a} = \|v_{p,1} - v_t\|$.

Under the assumption of a perfect perception module that will correctly detect all frontiers within its range, our planning approach will eventually visit every frontier becoming available, independent of the priority function. Since only graph nodes with a minimum coverage gain are considered frontier viewpoints, tours will not include already visited frontiers, guaranteeing that the robot always moves towards unexplored spaces. Therefore, our planner can ensure complete exploration of the environment.

V. EXPERIMENTS

A. Experimental Setup

Experiments are conducted in a Python-based 2D simulator with simplified sensing and navigation [21], [31]. Important aspects of the experiments are detailed below.

1) Scenario Setup: We use ProcThor [32] to sample multi-room indoor floorplans and realistic object placements with 4 different room categories (kitchen, bathroom, living room, bedroom). We generate environments with 3 kitchens, 3 bathrooms, 1 living room, and 1 bedroom, arranged with constrained connectivity (bedroom only accessible from the living room, bathrooms to the living room via the kitchens). We configure two scenario setups with a different target object and starting room type, detailed in the following sections. Top-down maps and object data are extracted for the simulator, and additional small objects are sampled to increase semantic feature density. Scenarios are curated to ensure challenging tasks where semantic features offer an advantage for target search. Both setups use 30 scenarios for generating intervention datasets and 34 scenarios for obtaining the evaluation results.

2) Oracle-based Data Generation: For training the semantic priority model, we generate synthetic interaction datasets by simulating expert interventions with an oracle model based on the expert model in Section IV-B. The oracle's priority model was designed and tuned to approximate a moderately rational expert providing occasional guidance while still allowing exploration. It assigns priorities based on room types, favoring frontiers in target rooms or exploring unseen rooms. Rooms are classified using a list of characteristic object classes for each room type. The expert model parameters (Section IV-B) used by the oracle are $\beta = 25.0$, $\rho = 0.0$, $\epsilon = 0.2$, and an intervention threshold $\tau = 0.05$. We also generate a dataset with an exponential distance discounting model, that is $\delta(f) = \exp\left(-\gamma d'_t(f)\right)$ with $\gamma = 0.1$, as well as datasets with varied β and τ to evaluate the robustness of our method to different expert behaviors (see Section V-C.3). Finally, we vary the number of episodes N_{eps} in the intervention dataset to evaluate the data efficiency of our method.

3) Training: The weights of the semantic priority model are trained using Adam [33] minimizing the negative log-likelihood of the observed expert choices (Eq. (6)) for 2000 epochs with learning rate 0.01. For each dataset, training uses 10 different random seeds, and the fixed expert model parameters are $\beta = 10.0,\ \rho = 0.1,\ \lambda = 0.7$.

B. Overview of Experiments

We evaluate the performance of our method in two task setups (see Section V-A) and present both qualitative and quantitative results. A coverage baseline, similar to [13], uses the planner proposed in Section IV-C, but with the priority function $P(f) = I(f)$. For both task setups, we first present qualitative results to illustrate an example scenario and the behavior of our method and the baseline. Second, we evaluate the target search performance of our method using quantitative metrics and compare it to different oracle methods, serving as upper bounds for the search performance. Using the same metrics, we evaluate the robustness of our method to different expert datasets by varying the number of interventions and parameters of the oracle model in the first scenario setup.

1) Metrics: We evaluate target search performance using the following metrics:

- *Path Length Ratio to Coverage (PLR)*: The episode-wise ratio of the path lengths l until target discovery between the compared semantic method l_{sem} and the coverage planner l_{cov}, i.e., PLR $= l_{\text{sem}}/l_{\text{cov}}$. The compared method reaches the target faster than coverage exploration for $PLR < 1$.
- *Success weighted by Path Length (SPL)*: The ratio of the traveled and shortest path to the target. A value of 1 indicates the shortest possible path to the target.

(a) Learned Priority Model
SPL = 0.784, PLR = 0.226

(b) Coverage Exploration
SPL = 0.177, PLR = 1.0

Fig. 2: Top-down views of an example scenario of the first task setup comparing coverage-driven exploration with our learned semantic priority model. Frontier nodes of the topological graph are colored black, and others are gray. Blue edges visualize the path taken by the robot; the larger blue circle is the robot's position at target discovery time, and the smaller blue circle is the initial position. The red rectangles are target objects. Object instances are visualized as small squares colored according to semantic class.

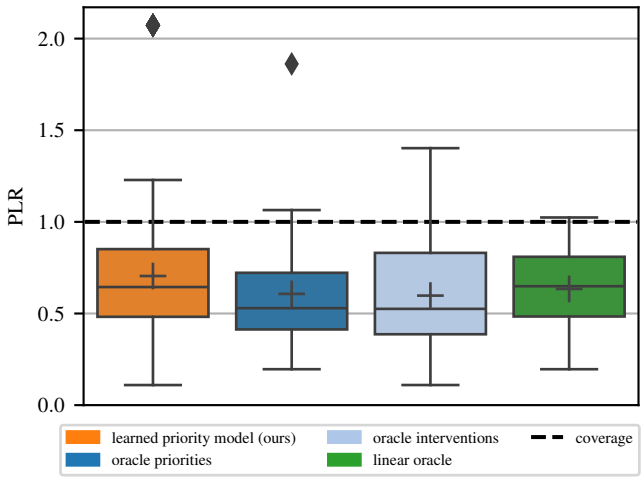

Fig. 3: Performance results in the primary target search task: Comparison of our method to oracle methods, displaying the episode-wise path length ratio (PLR) to the coverage baseline (dashed line) as boxplots.

While the SPL metric is common in object search [34], the PLR metric is proposed as the main metric to evaluate the efficiency of our method compared to the coverage planner as it quantifies the relative advantage over coverage per scenario.

2) Oracle Methods: In our performance evaluation, we compare our method to the following oracle methods:

- *Oracle Interventions* waypoint interventions from the oracle model overwrite the coverage baseline behavior
- *Oracle Priorities* guides the planner with the semantic priorities from the oracle model.
- *Linear Oracle* uses a linear expert model (as Eq. (2) with hand-tuned weights to obtain semantic priorities.

C. Primary Scenario Results

In the primary scenario setup, the target object is a bed in the bedroom, and the robot is initialized in one of the kitchens. Therefore locating the living room first and then the door to the bedroom is necessary.

1) Qualitative Results: Figure 2 compares the paths taken by the coverage planner and our target search planner with learned priorities in an example scenario (dataset $N_{eps} = 30$). The target object is in the bedroom (lower left) the robot starts in a kitchen (top right) and a large living room at the center connects the bedroom and kitchens. Figure 2 shows that our framework can guide the robot to the target object using a substantially shorter path than the coverage planner. Initially, the robot navigates to the living room instead of exploring the other doorway below, as observed objects in the living room are prioritized. The robot discovers a higher density of relevant objects in the lower part of the living room in turns in that direction. The robot also prioritizes door objects to search for the bedroom, leading the robot to the correct target room. Finally, discovered bedroom objects yield the highest priority and lead the robot to the target object. Conversely, the coverage planner prioritizes frontiers only based on coverage gain and first explores the large open spaces in the living room and, subsequently, the smaller rooms, ignoring semantic features. These exemplary results illustrate that our framework can leverage semantic features in the environment to achieve better target search efficiency than coverage-driven exploration.

TABLE I: Comparison of our method with the coverage baseline and oracle methods using the SPL metric defined in Section V-B.1, given as mean ± std.

Method	SPL (Task Setup 1)	SPL (Task Setup 2)
Coverage Priorities	0.406 ± 0.196	0.341 ± 0.313
Oracle Priorities	0.704 ± 0.202	0.564 ± 0.275
Oracle Intervention	0.712 ± 0.206	0.529 ± 0.281
Linear Oracle Priorities	0.650 ± 0.207	0.520 ± 0.271
Learned Priorities (ours)	**0.627 ± 0.225**	**0.520 ± 0.313**

2) Performance Results: We evaluate the target search performance of our method using $N_{eps} = 30$ in multiple test scenarios. The 340 episode results (10 training seeds and 34 test scenarios) are visualized as boxplot in Fig. 3. The orange boxplot shows that our method significantly outperforms the coverage planner (dashed line) in most scenarios (median PLR = 0.644), up to a best-case performance of PLR = 0.11. In 88% of episodes, our method is more efficient than the coverage planner, and in 97% of the episodes, PLR is smaller than 1.3, indicating that cases where our method misguides the robot are rare. Moreover, our approach matches the linear oracle and is only slightly outperformed by the non-linear oracle guidance. These results show that our approach learned the underlying semantic priorities of the oracle expert and effectively leverages them in multiple unseen scenarios. That is, by incorporating the learned priorities in the cost function of the planner, it prioritizes exploration frontiers likely to lead to the target. Table I additionally reports the SPL metric indicating a strong advantage in absolute target search performance over coverage exploration and competitive performance compared to oracle methods.

3) Robustness to Data Variation: Next, we analyze the robustness of our method to different dataset sizes N_{eps} and expert behavior by varying the oracle parameters. For each dataset, semantic priority models are trained and tested as described in Section V-C.2. Figure 4 shows the resulting PLR boxplots. The left subplot shows the results for a

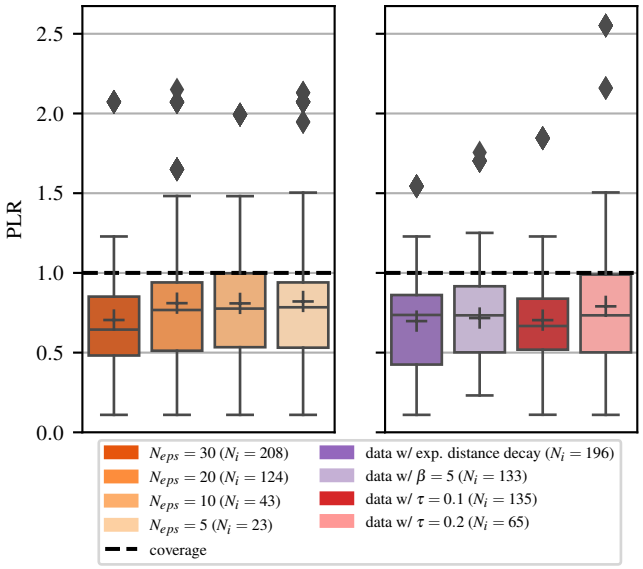

Fig. 4: Comparison of different dataset sizes and oracle behaviors used for training, displaying PLR performance of the resulting priority models.

reduced number of training episodes, ($N_{eps} = 30$ is the same as in Fig. 3). It is evident that with all 4 datasets, similar PLR performance is achieved. However, performance drops from $N_{eps} = 30$ to $N_{eps} = 20$, but further reduction up to $N_{eps} = 5$ does not affect the performance. Note that our method can achieve strong target search efficiency with only $N_i = 23$ expert interventions ($N_{eps} = 5$). A substantial improvement with more training data is only observed at $N_{eps} = 30$, which likely results from highly informative data points that only occur in this dataset, indicating that additional data can lead to further performance gains. The right subplot shows the results for 4 different oracle variations: exponential distance discounting instead of linear (Eq. (3)), reduced expert rationality β (increased noise, Eq. (5)), and increased expert intervention threshold (less engaged, more selective expert), all with $N_{eps} = 30$. Our method is robust to these changes and yields similar results across all variations. The lowest performance occurs for $\tau = 0.2$, since a less engaged expert might miss providing some informative interventions.

D. Secondary Scenario Results

The secondary scenario setup uses the same maps as the primary, but the target object is a toilet in one of the three bathrooms, and the robot starts in the living room. Here, the robot must first prioritize finding any of the kitchens that will lead to the bathrooms and the target object. In this setup it is harder to leverage semantic features as two kitchen-bathroom pairs might attract the robot but do not yield the target.

1) Qualitative Results: Figure 5 presents an example scenario of the secondary target search task, comparing the coverage planner with our planner guided by learned priorities. The target object is in the bathroom on the right side, and the robot starts in the bottom branch of the living room. The living room connects to a large bedroom in the center and 3 kitchen-bathroom pairs at the top of the map. The coverage robot incurs much performance loss when exploring the bedroom,

(a) Learned Priority Model
SPL = 0.509, PLR = 0.530

(b) Coverage Exploration
SPL = 0.270, PLR = 1.0

Fig. 5: Top-down views of an example scenario of the second task setup comparing the behavior of coverage-driven exploration and our learned semantic priority model. Visuals follow the same conventions as in Fig. 2.

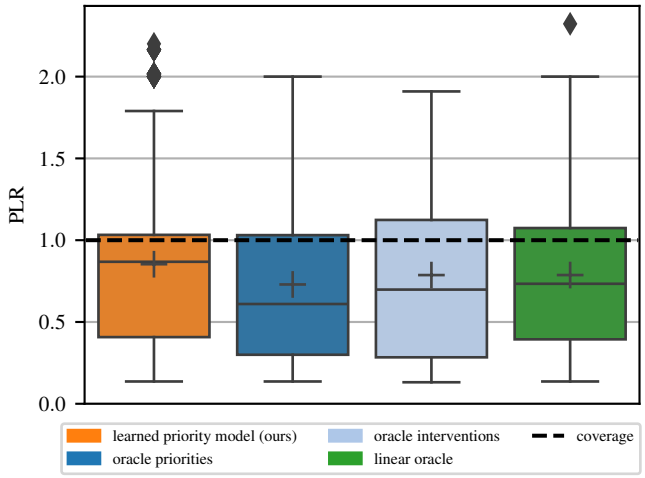

Fig. 6: Performance results in the secondary target search task: Comparison of our method to oracle methods, displaying the episode-wise path length ratio (PLR) to the coverage baseline as boxplots.

while the learned semantic priorities favor continuing in the living room. Both remaining paths in the upper part of the map are very similar, as the semantic features cannot strongly favor one direction over the other; all small rooms are semantically promising. This example scenario indicates that the advantage of semantic over coverage exploration is less pronounced in this scenario setup, as only the bedroom is a clearly semantically irrelevant area, while the remaining rooms are all prioritized.

2) Performance Results: Quantitative performance results in the secondary task setup are presented in Fig. 6, analogous to Section V-C.2. While our method outperforms the coverage planner (PLR < 1) in most episodes, the mean PLR of 0.853 is closer to 1 than in the primary task setup. This indicates more similar behavior of our method to the coverage planner, possibly as semantic priorities are less informative for target search. This is also supported by the PLR boxplots of the oracle methods, showing that more episodes perform similar to coverage than in the primary setup. Moreover, this task setup features a larger median gap between our approach and the oracle methods. This shows that the difficulty of this task setup is exacerbated when using potentially noisy learned

semantic priorities, giving more influence to the coverage gains in the tour cost function (Eq. (8)). However, while some scenarios do not provide much room for improvement over coverage, the results show that our approach substantially improves target search efficiency in many other scenarios.

VI. Conclusion

In this paper, we presented a novel approach to target search in unknown environments, combining semantic priorities learned from expert guidance with a global exploration planner. We trained the semantic priority model weighting exploration frontiers based on semantic features, such that a derived expert model matches a dataset of expert interventions. The combinatorial exploration planner prioritizes frontiers based on semantic priority and expected coverage gain, ensuring robust exploration independent of the learned model. The results show that the exploration planner guided by the learned priority model exhibits efficient target search behavior and outperforms a purely coverage-driven planner variant across different scenarios and simulated expert datasets. Future work will consider more realistic environments with complex semantic relationships and learning from real human data.

References

[1] D. S. Chaplot, D. Gandhi, A. Gupta, and R. Salakhutdinov, "Object Goal Navigation using Goal-Oriented Semantic Exploration," *Advances in Neural Information Processing Systems*, 2020.

[2] N. Kim, O. Kwon, H. Yoo, Y. Choi, J. Park, and S. Oh, "Topological Semantic Graph Memory for Image-Goal Navigation," in *6th Annual Conference on Robot Learning*, 2022.

[3] 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)*, 2022.

[4] M. Hahn, D. S. Chaplot, S. Tulsiani, M. Mukadam, J. M. Rehg, and A. Gupta, "No RL, No Simulation: Learning to Navigate without Navigating," in *Advances in Neural Information Processing Systems*, 2021.

[5] A. Majumdar, G. Aggarwal, B. S. Devnani, J. Hoffman, and D. Batra, "ZSON: Zero-Shot Object-Goal Navigation using Multimodal Goal Embeddings," in *Advances in Neural Information Processing Systems*, 2022.

[6] J. Chen, G. Li, S. Kumar, B. Ghanem, and F. Yu, "How To Not Train Your Dragon: Training-free Embodied Object Goal Navigation with Semantic Frontiers," in *Robotics: Science and Systems XIX*, 2023.

[7] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "Vlfm: Vision-language frontier maps for zero-shot semantic navigation," in *International Conference on Robotics and Automation (ICRA)*, 2024.

[8] M. F. Ginting, S.-K. Kim, D. D. Fan, M. Palieri, M. J. Kochenderfer, and A.-a. Agha-Mohammadi, "SEEK: Semantic Reasoning for Object Goal Navigation in Real World Inspection Tasks," 2024.

[9] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson Env: Real-World Perception for Embodied Agents," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[10] 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 *2017 International Conference on 3D Vision (3DV)*, 2017.

[11] C. Cao, H. Zhu, H. Choset, and J. Zhang, "TARE: A Hierarchical Framework for Efficiently Exploring Complex 3D Environments," in *Robotics: Science and Systems XVII*. Robotics: Science and Systems Foundation, 2021.

[12] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "FUEL: Fast UAV Exploration Using Incremental Frontier Structure and Hierarchical Planning," *IEEE Robotics and Automation Letters*, 2021.

[13] J. Huang, B. Zhou, Z. Fan, Y. Zhu, Y. Jie, L. Li, and H. Cheng, "FAEL: Fast Autonomous Exploration for Large-scale Environments With a Mobile Robot," *IEEE Robotics and Automation Letters*, 2023.

[14] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep Reinforcement Learning from Human Preferences," in *Advances in Neural Information Processing Systems*, 2017.

[15] J. Abramson, A. Ahuja, F. Carnevale, P. Georgiev, A. Goldin, A. Hung, J. Landon, J. Lhotka, T. Lillicrap, A. Muldal, G. Powell, A. Santoro, G. Scully, S. Srivastava, T. von Glehn, G. Wayne, N. Wong, C. Yan, and R. Zhu, "Improving Multimodal Interactive Agents with Reinforcement Learning from Human Feedback," 2022.

[16] N. Wilde, D. Kulic, and S. L. Smith, "Active Preference Learning using Maximum Regret," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[17] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee, "Transporter Networks: Rearranging the Visual World for Robotic Manipulation," in *Proceedings of the 2020 Conference on Robot Learning*, 2021.

[18] J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S. Srinivasa, "Expert Intervention Learning," *Autonomous Robots*, 2022.

[19] Z. Meng, H. Qin, Z. Chen, X. Chen, H. Sun, F. Lin, and M. H. Ang, "A Two-Stage Optimized Next-View Planning Framework for 3-D Unknown Environment Exploration, and Structural Reconstruction," *IEEE Robotics and Automation Letters*, 2017.

[20] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep Reinforcement Learning Robot for Search and Rescue Applications: Exploration in Unknown Cluttered Environments," *IEEE Robotics and Automation Letters*, 2019.

[21] M. Lodel, B. Brito, Á. Serra-Gómez, L. Ferranti, R. Babuška, and J. Alonso-Mora, "Where to Look Next: Learning Viewpoint Recommendations for Informative Trajectory Planning," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022.

[22] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97*, 1997.

[23] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A Real-time Spatial Perception System for 3D Scene Graph Construction and Optimization," in *Robotics: Science and Systems XVIII*, 2022.

[24] D. Sadigh, A. Dragan, S. Sastry, and S. Seshia, "Active Preference-Based Learning of Reward Functions," in *Robotics: Science and Systems XIII*, 2017.

[25] R. A. Bradley and M. E. Terry, "Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons," *Biometrika*, 1952.

[26] A. Blum, P. Chalasani, D. Coppersmith, B. Pulleyblank, P. Raghavan, and M. Sudan, "The minimum latency problem," in *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing - STOC '94*, 1994.

[27] D. Pisinger and S. Ropke, "Large Neighborhood Search," in *Handbook of Metaheuristics*, M. Gendreau and J.-Y. Potvin, Eds., 2019.

[28] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis, II, "An Analysis of Several Heuristics for the Traveling Salesman Problem," *SIAM Journal on Computing*, 1977.

[29] G. A. Croes, "A Method for Solving Traveling-Salesman Problems," *Operations Research*, 1958.

[30] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, 1968.

[31] M. Everett, Y. F. Chen, and J. P. How, "Motion Planning Among Dynamic, Decision-Making Agents with Deep Reinforcement Learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[32] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, K. Ehsani, J. Salvador, W. Han, E. Kolve, A. Kembhavi, and R. Mottaghi, "ProcTHOR: Large-Scale Embodied AI Using Procedural Generation," in *Advances in Neural Information Processing Systems*, 2022.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR*, 2015.

[34] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, and A. R. Zamir, "On Evaluation of Embodied Navigation Agents," 2018.