2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

Open Vocabulary Object Search Utilizing Large Language Models and Fuzzy Inferencing

Akash Chikhalikar, Ankit A. Ravankar, Jose Victorio Salazar Luces, and Yasuhisa Hirata

***Abstract*— Open vocabulary task execution is crucial in autonomous robotics, particularly for indoor service robots operating in dynamic, human-centric environments. Conventional dictionary-based approaches either fail to capture the diversity in interactions between objects and humans or often face scalability issues in memory and computation over time. Thus, a framework capable of executing high-level tasks and robust *open-set* capabilities is desirable. We consider the task of searching for dynamic objects in an indoor environment called Object Search. While the state-of-the-art approaches focus on the most effective ways to search for a closed set of objects, we propose a framework capable of generalizing to unknown, unseen, and ultimately an open set of objects. Our framework consists of a method to leverage priors of a fixed set of objects to generate *task-driven priors* for an open set of objects. We utilize Large Language Models (LLMs) and fuzzy logic to facilitate this prior generation. Additionally, the proposed framework also captures the physical layout of the environment to inform task-driven prior generation. Finally, we validate our framework through extensive real-world experiments and provide comparisons with competitive methods, demonstrating its effectiveness in generalizing to an open-set of objects. The results demonstrate our framework's superiority in reducing search time, distance, and number of visited landmarks, outperforming related methods.**

***Index Terms*— Open Vocabulary, Object Search, Fuzzy Inferencing, Large Language Models.**

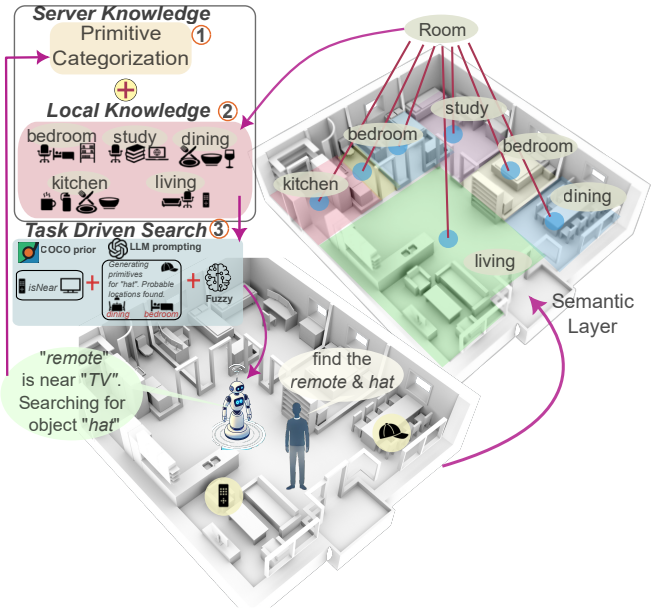


Fig. 1: Illustration of our approach for *open-vocabulary* object search, leveraging *task-driven priors* generated from local knowledge and primitive-based categorization.

I. Introduction

Open Vocabulary refers to a system's ability to understand, interpret, and respond to a wide range of natural language inputs without being constrained to predefined commands. This flexibility is key to creating more intuitive and accessible human-robot interactions, allowing the robots to function effectively in dynamic, unstructured environments. Despite the advances in closed-set object search, the unpredictable nature of indoor environments, where new objects are frequently introduced, necessitates an open-vocabulary approach. Open Vocabulary systems enable robots to comprehend and act on instructions phrased in diverse ways, accommodating the natural variability in human communication. This capability is particularly important in environments where users may lack technical expertise or instructions vary significantly, such as in home automation, healthcare, and customer service. For example, a robot equipped with open vocabulary capabilities can understand the similarity between requests like *"Please bring me a glass of water"*, *"Can you get me some water?"* or *"I need a drink of water,"* and execute the same set of actions for each command. Recently, Large Language Models (LLMs) have been shown to play a central role in interpreting and planning open vocabulary commands. However, the generalized capabilities of an LLM should be combined with environment-specific information for proper execution, as shown in Fig. 1.

Generative Pre-trained Transformer (GPT) models, a type of LLM, have shown remarkable potential for generalization through few-shot learning [1]. These GPT-in-the-loop systems have been applied in various domains, including multi-robot coordination [2], long-horizon planning [3], and human-robot collaboration [4].

GPTs can also be utilized for the high-level task of Object Search, which is essential for service robots performing various tasks (e.g., "tidy up my room") in indoor environments. An indoor setting can be viewed as a network of rooms, landmarks, and objects. While the landmarks (e.g., fridge, bed) tend to remain stationary over time, objects (e.g., bottle, bag) are frequently moved, often multiple times within a single day, leading to highly uncertain and dynamic positions. Conducting an exhaustive or coverage-based search for such objects is inefficient and time-consuming. A more effective approach is to use a probabilistic prior to

The authors are with the Graduate School of Engineering, Department of Robotics, Tohoku University, Sendai 980-8579, Japan. (e-mail: {a.k.chikhalikar, ankit, j.salazar, hirata}@srd.mech.tohoku.ac.jp)

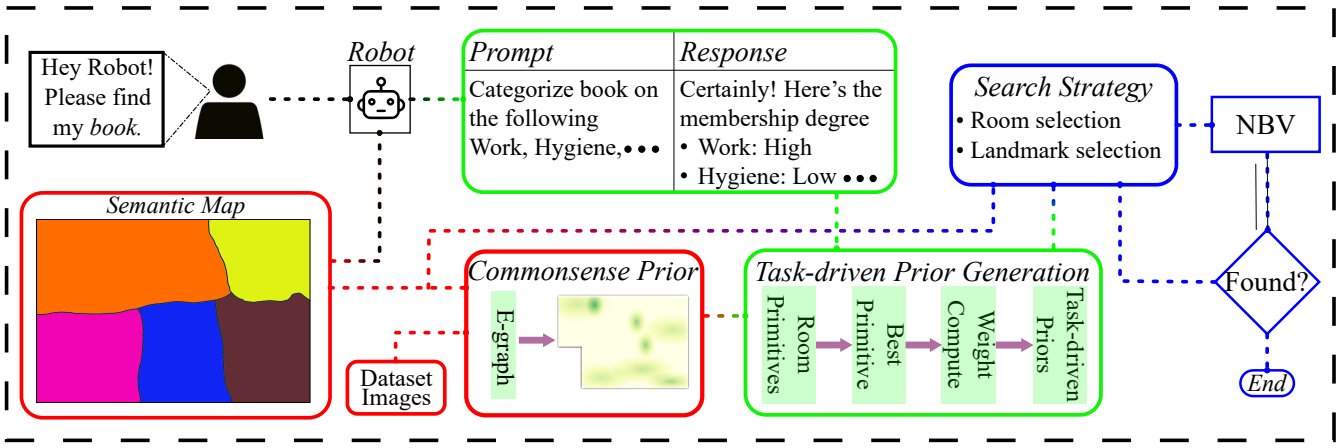


Fig. 2: Overview of our framework. We actively query LLMs and use fuzzy logic for *task-driven prior generation*. For more details, please refer to Sec. IV-A (common-sense prior), Sec. IV-B (task-driven prior) and Sec. IV-C (search strategy).

narrow down the search space [5], [6]. If these priors are generated based on the task rather than stored in memory, the approach can be scaled for any number of objects. A decision-making strategy then employs these priors to guide the robot toward candidate viewpoints, making the success of the search heavily dependent on both the quality of the prior and the decision-making strategy.

In our approach, the priors are generated based on the assumption that the objects are located on or within a fixed set of landmarks. This limits the search space to areas on or around these landmarks, which covers most real-world scenarios since objects are typically placed on (e.g., tables, beds, sofas) or inside (e.g., fridges, closets) landmarks. Importantly, we do not restrict the number of objects the robot may be tasked to search in the environment. This is important because, in real-world scenarios, new objects are frequently introduced. It is infeasible to store the priors for all existing and future items. While GPTs can aid in planning, it is crucial to integrate their input with personalized knowledge of the environment. Thus, generating task-driven priors is essential for effectively executing open vocabulary object search.

The main contributions of this paper include:

- A novel framework that extends object search into the *open vocabulary* domain by integrating fuzzy logic and LLMs, significantly expanding the scope and applicability of object search.
- A novel strategy for *task-driven prior generation* that considers the environment's topology and primitive attributes to guide the search process.
- Comprehensive experimental results in real-world environments, along with benchmarking existing state-of-the-art methods.

An overview of our framework is shown in Fig. 2. The rest of the paper is structured as follows. Section II reviews related work and distinguishes our approach from current methods. Section III outlines the preliminaries for open vocabulary object search. Section IV, details our framework for generating task-driven priors, which is the core contribution of our work. Section V presents the experimental studies and results. Finally, Section VI offers a conclusion and discussion.

II. Related Works

Object search has been extensively studied in robotics, particularly for service robotics scenarios using various approaches ([5]–[13]). Among these, several methods [5], [6], [7], [8], and [12] treat object search as a Next Best View (NBV) problem. Meanwhile, other methods such as [9], [10], and [11] model object search as a Partially Observable Markov Decision Process (POMDP).

In [6], the authors propose a coverage algorithm wherein the NBV is selected by maximizing the belief around a given point. [7] leveraged the commonsense knowledge base combined with on-site learning to determine the NBV. This knowledge base is distilled into a scene graph and used for target localization in [12]. In [8], the robot is equipped with a pan-tilt camera, and a method is developed to optimize both the viewing position and angle to enhance the chances of locating the object.

In contrast to NBV-based approaches, [9] presents a multi-resolution POMDP framework that optimizes joint rotations to maximize visual area coverage. [10] introduces a spatial correlation model to optimize the POMDP-based object search, while [11] proposes a POMDP framework that accounts for clutter in the environment during the search process.

Despite their differences, a key commonality across these methods is that they operate within a closed set of objects. In contrast, our framework addresses open vocabulary object search, allowing for generalization to an open set of objects. Additionally, the priors used in existing methods are typically pre-determined and stored in memory. For an open-set, this approach is impractical due to the large memory requirements. Our method of *task-driven prior generation* overcomes this challenge by storing priors for only a limited number of objects, thus enabling a more flexible and

Reference	Scope	LLM-based	Prior
[7]	Closed-set	-	Preset
[8]	Closed-set	-	Preset
[9]	Closed-set	-	Preset
[14]	Closed-set	-	Preset
[12]	Closed-set	✓	Preset
Ours	Open-set	✓	Task-driven

TABLE I: Comparison of our work with related works.

memory-efficient search. A summary of these differences is presented in Table III.

III. Preliminaries

Object Search is a high-level task that relies on an underlying mapping and reasoning framework. To support its execution, we define several key preliminary concepts.

A. Semantic Map

A semantic map extends beyond the conventional grid-based map by incorporating information necessary for semantic reasoning and efficient search space pruning. Our framework considers the environment a hierarchical structure consisting of rooms, regions, and objects. Therefore, any method capable of generating such hierarchical semantic maps ([14]–[20]) can be integrated into our system. To create this semantic layer, we require an object detection module, a data association module, and a localization module:

1) *Object Detection*: We use Yolov7 [21] for object detection, which provides real-time, high-accuracy recognition of objects in the environment.
2) *Data Association*: To maintain consistency in tracking objects, we employ the Hungarian algorithm [22], which effectively associates detected objects with existing entries in the map.
3) *Localization*: Kalman filtering is used for estimating the positions of objects over time, improving the robustness of localization in dynamic environments.

The information gathered from these modules is then integrated into an octomap, which we generate using an RGB-D SLAM technique like RTABMap [23]. The octomap allows us to represent the 3D structure of the environment, including free and occupied spaces.

After constructing the octomap, we use Voronoi graphs to segment the environment geometrically into distinct regions. These regions are then labeled based on empirical data from the Places365 dataset [24], allowing the map to encode room-related, landmark-related, and occupancy-related information. This hierarchical semantic mapping approach enables the robot to reason about the environment more effectively, focusing the search on likely locations for objects.

For further implementation details, we refer the reader to our previous work [14], where we explain the full process of semantic map generation and integration into the object search task.

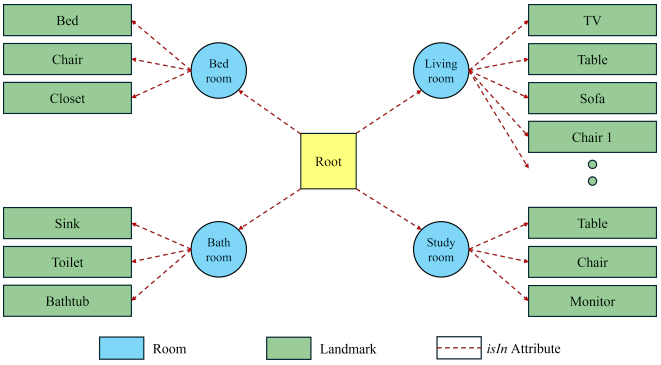


Fig. 3: Partial E-graph of the environment.

B. Environment Graph

We utilize the generated semantic map to extract the environment graph (E-graph), a fully connected, one-way graph with a 3-layer hierarchical structure representing rooms and the landmarks within the environment. The E- graph is denoted as $G = \{V, E\}$, where V represents the vertices with attributes $\{Room(R), Landmark(L)\}$, and E represents the edges with type $\{isIn, isNear\}$. The $\{isIn\}$ attribute is binary, indicating whether a child node (e.g., a landmark) is contained with a parent node (e.g., a room). The $\{isNear\}$ attribute quantifies the proximity between nodes, calculated using the Euclidean distance between them.

The graph's hierarchical structure is as follows:

1) *First Layer*: The root node represents the entire indoor area.
2) *Second Layer*: This layer consists of nodes representing individual rooms within the environment.
3) *Third Layer*: Nodes in this layer represent landmarks located within or near the rooms.

All nodes directly connected to the root node (the indoor area) have their $isIn$ attribute set to 1. For nodes representing rooms, the $isNear$ attribute for their edges connecting to landmark nodes is assigned only if the $isIn$ attribute is zero, indicating that the landmark is not within the room but in proximity to it. The values for these attributes are computed based on their geometric positions in the semantic map using Euclidean distance. The E-graph's hierarchical, attribute-rich structure enables efficient reasoning about the environment's layout. A partial representation of the E-graph is illustrated in Figure 3.

IV. Methodology

We leverage the semantic map, E-graph, and fuzzy logic for our open vocabulary object search. We start by extracting common-sense priors from publicly available datasets for a limited set of objects. These objects are then categorized based on their degree of membership to various primitive attributes. Using these attributes, we generate task-driven priors that guide the object search, allowing the framework to generalize effectively to an open vocabulary of objects.

A. Common-sense Prior Generation

As discussed earlier in our framework, we leverage primitive attributes and priors derived from a fixed set of objects. However, obtaining these priors for a fixed set of objects is challenging. For this purpose, we use the popular COCO dataset [25], considering object-landmark co-occurrences and the spatial topology of the environment. Our framework includes two types of priors:

- Object $isOn$ landmark
- Object $isNear$ landmark

We exclude the 'object $isIn$ landmark' prior (e.g., milk in the fridge) since we assume that objects are placed in visually accessible locations. These priors were also utilized in our previous work [26] [27] and the current study extends this by incorporating the E-graph for a more holistic consideration of spatial topology.

The prior for 'object $isOn$ landmark' ($P_{On}(O|L)$) is directly calculated from the instance masks in the COCO dataset. On the other hand, the prior for 'object $isNear$ landmark' requires a deeper analysis of the spatial topology. We first mine the COCO dataset to identify co-occurrences between objects (O) and landmarks (L), filtering for indoor images. The filtered images are then scanned for the presence for both the object and landmark within the same frame, using ground truth annotations. The co-occurrence prior is estimated as follows:

$$P_{Corr}(O|L) = \frac{N(O \cap L) + \nu}{N(O) + \nu d} \tag{1}$$

Here, $N(\cdot)$ represents the count of observations in the COCO dataset, d is the number of classes in the dataset, and ν is the additive smoothing parameter, set to $\nu = 0.5$ based on Lidstone's law.

The $isNear$ prior is derived from the co-occurrence prior and the spatial topology of the environment. We use the Spatial Decay Function (SDF) from [26] to incorporate spatial topology. The Euclidean distance between different landmark pairs ($d(L, L')$) is calculated. The SDF value is obtained using a parabolic decay function, which outputs a value of 1 for $d(L, L^{'}) < 1$ and 0 for $d(L, L^{'}) > 3$. This value is applicable only when both the landmarks belong to the same room (i.e., the edges with the {isIn} attribute from both landmarks point to the same room node).

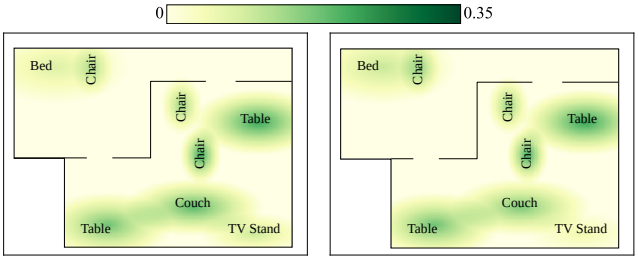
$$P_{Near}(O|L) = \sum_{\forall L' \neq L} z(L, L') SDF(L, L') P_{Corr}(O|L) \tag{2}$$

$$z(L, L') = \begin{cases} 1 & \text{if } L, L' \; isIn \; R \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The overall common-sense prior is obtained by linearly combining the *isOn* and *isNear* priors.

$$P_{CSP}(O|L) = \eta \{P_{On}(O|L) + \lambda P_{Near}(O|L)\} \tag{4}$$

where, η is a normalization constant and $\lambda (= 0.5)$ is the hyperparameter for recombination. The priors obtained using this framework for objects like phones and backpacks are shown in Figure 4. Similar priors are generated for objects such as cups, remotes, books, and toothbrushes. These objects collectively form the set of dictionary objects ($\mathbb{O}_D$), with each object in this set denoted by D.



(a) $P_{CSP}(phone|L)$ (b) $P_{CSP}(backpack|L)$

Fig. 4: Commonsense prior for phone and backpack.

B. Task-driven Prior Generation

Our framework generates a unique prior for each target object by combining the common-sense priors (P_{CSP}) of dictionary objects into a task-driven prior (P_{TDP}). This prior is calculated as follows:

For $D \in \mathbb{O}_D$:

$$P_{TDP}(T|L) = \zeta \left\{ \sum_{\forall L \in R} \sum_{\forall D, R} \alpha_{D,R} P_{CSP}(D|L) \right\} \tag{5}$$

where, ζ is the normalization constant, and $\alpha_{D,R}$ are the weights for each commonsense prior.

To determine these weights, we use concepts from fuzzy logic [28], categorizing objects based on primitive attributes. In the context of an indoor environment, an object's purpose serves as its primitive attribute. Subsequently, the purpose-based primitive p belongs to the set $\mathbb{P}$ = {Entertainment, Work, Hygiene, Consumable and Essentials}. The degree of membership for each object in each primitive $M_p(O)$ belongs to the linguistic fuzzy set $\mathbb{F}$ = {LOW, MEDIUM, HIGH}.

When the robot is tasked with finding an object, it first queries a LLM to obtain the target's membership degree in each primitive. Similarly, the membership degrees of dictionary objects (i.e., phone, remote) are retrieved by querying the LLM. The fuzzy set $\mathbb{F}$ is then used to represent the similarity degree between the target and dictionary object for every primitive. The similarity matrix, ($S_p(T, D)$) is structured as follows:

$M_p(T)$ \ $M_p(D)$	LOW	MEDIUM	HIGH
LOW	HIGH	MEDIUM	LOW
MEDIUM	MEDIUM	HIGH	MEDIUM
HIGH	LOW	MEDIUM	HIGH

TABLE II: $S_p(T, D)$ for any primitive $p \in \mathbb{P}$.

For example, if a target object T has a LOW degree of membership in the primitive p = Entertainment, the similarity outcomes for a dictionary object D are:

- If $M_p(D) = \text{HIGH} \implies S_p(T, D) = \text{LOW}$
- If $M_p(D) = \text{MEDIUM} \implies S_p(T, D) = \text{MEDIUM}$
- If $M_p(D) = \text{LOW} \implies S_p(T, D) = \text{HIGH}$

Since the target object may have high similarity with multiple dictionary objects across different primitives, we also need to resolve the weights assigned to each primitive for a given room. To achieve this, we consider the similarity magnitude between objects and the room's degree of membership in each primitive. For example, a study room is more associated with the Work primitive, while a bathroom is linked to Hygiene. Therefore, weights for objects differ depending on the room's context.

If a room has a high membership degree for a particular primitive, the similarity values $S_p(T, D)$ for that primitive take precedence over others. If a room shows equal membership degrees for multiple primitives, all receive equal precedence. Simply put, in the first step (Eq. 6) of this hierarchical logic , the most significant primitive(s) for each room are decided and in the second step (Eq. 7), the weights are decided based on the degree of similarity in that primitive. The selection of the most significant primitive(s) and the determination of weights is as follows:

$$\begin{aligned} &\forall\ R \in G\ \forall\ L\ s.t\ L\ isIn\ R\text{:} \\ &\quad p^* : M_{p^*}(R) \geq M_{p'}(R) \qquad \forall\ p^*, p' \in \mathbb{P} \\ &\quad D^* : S_{p^*}(T, D^*) \geq S_{p^*}(T, D^{'}) \quad \forall\ D^*, D' \in \mathbb{O}_D \end{aligned} \tag{6}$$

$$\alpha_{D,R} = \begin{cases} \dfrac{\sum_{\forall p, D} S_p(T, D)}{||p^*|| + ||D^*||} & if\ p = p^*\ and\ D = D^* \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

The weights $\alpha_{D,R}$ are then substituted into Equation 5 to generate the task-driven priors, which are subsequently normalized to complete the process.

C. Object Search Strategy

The robot executes the search using a hierarchical strategy guided by the E-graph. First, the robot identifies the room with the highest probability of containing the target object by summing the task-driven priors for all landmarks within each room. Once the most likely room is determined, the robot visits the landmark within that room with the highest prior. If the object is not found there, it proceeds to other landmarks in the room, considering a trade-off between distance and the task-driven prior. After searching all landmarks in the current room, the robot moves to the next most probable room. This two-tiered approach minimizes unnecessary transitions between rooms, mimicking a human-like search strategy. The complete hierarchical strategy is detailed in Algorithm 1.

The algorithm first determines the most probable room to search based on task-driven priors. Within that room, it prioritizes landmarks by considering their prior probability and proximity, mimicking a human-like search strategy. If the object is not found, the process iterates through other rooms in descending order of probability, ensuring an efficient search path.

Algorithm 1 Algorithm for open vocabulary object search

Require: Semantic Map: $\mathcal{M}$, E-graph: G
Input: $Object : O,\ Prior : P_{TDP}(O|L)$
while Object Not_Found **do**
  for each $R \in G$ **do**
    for each $L \in G\ s.t.\ E(R, L) = isIn$ **do**
      $P_{TDP}(O|R) \leftarrow +P_{TDP}(O|L)$
    end for
  end for
  $R^* \leftarrow \arg\max_R(P_{TDP}(O|R))$
  for each $L \in G\ s.t.\ E(R^*, L) = isIn$ **do**
    Calc: $(Dist./Prior)$ ▷ Trade-off
  end for
  $NBV \leftarrow \arg\max_L\ (Dist./Prior)$
end while

V. Experimental Studies

All experiments were conducted at the Aobayama Living Lab at Tohoku University. Established under the Japanese Government's Moonshot R&D program, the living lab is designed to accelerate research in service robots for long-term care and support the aging population by providing a test bed for defining and implementing robot hardware, novel algorithms, and empirical tests in a simulated setting. The facility simulates a home environment, featuring different rooms (e.g., living room, bathroom, bedroom) and household objects (e.g., bed, table, chair). A digital twin of this environment has also been developed to facilitate collaborative research [29], [30]. The lab is equipped with various sensors [31], motion capture systems, and robots, all of which can be controlled using IoT-based voice assistance services such as Alexa, Google Assistant, or HomeAssistant. The environment is shown in Fig. 5 below.

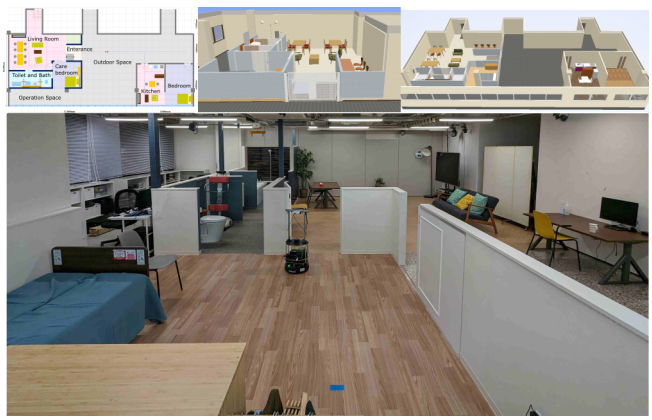


Fig. 5: Indoor testing environment

A. Hardware

We used the Locobot WX250 6DoF as the mobile base for the experiments. The Locobot is equipped with an RGB-D camera (Intel Realsense D-435), an onboard computer (Intel NUC Mini PC) and a 2D Lidar (RPLIDAR A2). The object-detection, navigation stack, and the OpenAI API were executed on a server PC featuring an i9-12900K processor and RTX-3090 graphics card. Implementing the mapping and navigation stack and communication between the robot and the server was managed using the ROS open-source framework [32].

B. Comparisons

We compared our framework against the following methods:

- *Distance-TSP (D-TSP)*: This method formulates object search as a Travelling Salesperson Problem (TSP) to determine the optimal sequence of locations to visit, focusing solely on minimizing travel distance. The TSP is solved using Google-OR Tools' Routing solver [33].
- *Closest cosine neighbor search (CNS)*: This approach calculates the cosine similarity between the target objects and dictionary objects using GloVe embeddings [34]. The prior of the dictionary object with the highest similarity to the target is selected. The search strategy follows the process outlined in Section IV-C.
- *Naive-LLM (n-LLM)*: A Large Language Model(LLM) is prompted with the target object and provides the spatial structure of the environment as a graph, along with the robot's current location. The LLM then suggests the sequence of locations to visit.

C. Evaluation Metrics

The methods were evaluated using the following metrics:

- *Distance (Dist.)*: The total distance traversed by the robot during the search.
- *Time (T)*: The time taken by the robot to locate the object.
- *Landmark (Ldmk.)*: The number of landmarks visited by the robot during the search.
- *Probability Weighted Success (PWS)*: For a total of N runs, where object locations vary with probabilities P_i and corresponding search distances D_i, PWS is defined as:

$$PWS = \frac{\sum_{i=1}^{N} P_i(O|L) D_i}{N} \tag{8}$$

- *Success by Path Length (SPL)*: This metric measures the robot's path efficiency compared to an optimal (oracle) agent that follows the shortest possible path. SPL metric is defined as follows:

$$SPL = \frac{1}{N} \sum_{i=1}^{N} \frac{D_{opt}}{max(D, D_{opt})} \tag{9}$$

where, D_{opt} is the optimal distance to the object, and D is the actual path length taken by the robot.

D. Experiment setup

The robot was tasked with searching for three new objects (laptop, bottle, and hair dryer), starting from the same position for each method. The objects were placed in the three most likely locations based on the COCO dataset. The average results of the experiments are summarized in Table III.

Method	Metrics (Avg.)				
	Time(s) ↓	Ldmk. ↓	Dist.(m) ↓	PWS(m) ↓	SPL ↑
D-TSP	75.67	4.95	12.31	4.64	0.31
CNS	66.89	4.12	10.18	4.4	0.44
n-LLM	61.33	**3.54**	11.65	3.8	0.53
Ours	**60.61**	3.88	**9.76**	**3.55**	**0.57**

TABLE III: Comparison of our framework with other approaches

As shown in Table III, our framework outperforms the other methods across most metrics. It achieves the shortest average time (60.61s), travels the least distance (9.76 units), and records the lowest Probability Weighted Success (PWS) score (3.55). Additionally, it achieves the highest Success by Path Length (SPL) of 0.57, indicating a more efficient search strategy. Although n-LLM visits slightly fewer landmarks on average, our method demonstrates a better overall balance of speed, efficiency, and accuracy in open vocabulary object search.

VI. Conclusion

We presented a novel framework for open vocabulary object search in indoor environments, leveraging spatial topology and fuzzy logic for more effective task-driven prior generation. Our framework integrated the semantic map and environment graph (E-graph) with common-sense priors derived from a limited set of objects, to generalize to an open-set of objects, effectively guiding the robot to potential locations. The experimental results demonstrate that our approach not only achieves faster search times and reduced travel distances but also maintains high efficiency, outperforming other related approaches. A key strength of our framework lies in its ability to adapt to dynamic environments despite having information for only a few objects. Using task-driven priors and a hierarchical search strategy mirrors human-like reasoning in object search, minimizing unnecessary transitions and optimizing both time and effort.

However, a current limitation is using a closed set of primitives. While these primitives allow the framework to categorize and search for various objects, certain items (e.g., jewelry) with unique or less common purposes may not fit into the existing categories, potentially limiting the framework's open-vocabulary capabilities. Future work should aim to identify a more extensive and flexible set of primitives that can exhaustively categorize any object.

In summary, our framework not only bridges the gap in open vocabulary object search by introducing task-driven priors but also establishes a novel use of fuzzy logic for object search. By effectively handling new and unseen objects in

dynamic environments, our approach significantly advances the field of service robotics.

ACKNOWLEDGMENT

This work was partially supported by JST Moonshot R&D [Grant Number JPMJMS2034], JSPS Kakenhi [Grant Number JP21K14115 and JP24K07399], and JST SPRING [Grant Number JPMJSP2114].

REFERENCES

[1] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," *Autonomous Robots*, vol. 47, no. 8, pp. 1087–1102, 2023.

[2] B. Yu, H. Kasaei, and M. Cao, "Co-navgpt: Multi-robot cooperative visual semantic navigation using large language models," 2023.

[3] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, "Sayplan: Grounding large language models using 3d scene graphs for scalable task planning," in *7th Annual Conference on Robot Learning*, 2023.

[4] S. Izquierdo-Badiola, G. Canal, C. Rizzo, and G. Alenyà, "Plancollabnl: Leveraging large language models for adaptive plan generation in human-robot collaboration," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 17344–17350, 2024.

[5] T. Kollar and N. Roy, "Utilizing object-object and object-scene context when planning to find things," in *2009 IEEE International Conference on Robotics and Automation*, pp. 2168–2173, 2009.

[6] A. C. Hernandez, E. Derner, C. Gomez, R. Barber, and R. Babuška, "Efficient object search through probability-based viewpoint selection," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6172–6179, 2020.

[7] S. Hasegawa, A. Taniguchi, Y. Hagiwara, L. El Hafi, and T. Taniguchi, "Inferring place-object relationships by integrating probabilistic logic and multimodal spatial concepts," in *2023 IEEE/SICE International Symposium on System Integration (SII)*, pp. 1–8, 2023.

[8] Y. Zhang, G. Tian, X. Shao, S. Liu, M. Zhang, and P. Duan, "Building metric-topological map to efficient object search for mobile robot," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 7, pp. 7076–7087, 2022.

[9] K. Zheng, Y. Sung, G. Konidaris, and S. Tellex, "Multi-resolution pomdp planning for multi-object search in 3d," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2022–2029, 2021.

[10] K. Zheng, R. Chitnis, Y. Sung, G. Konidaris, and S. Tellex, "Towards optimal correlational object search," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 7313–7319, 2022.

[11] Y. Chen and H. Kurniawati, "Pomdp planning for object search in partially unknown environment," in *Advances in Neural Information Processing Systems* (A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, eds.), vol. 36, pp. 53146–53157, Curran Associates, Inc., 2023.

[12] W. Ge, C. Tang, and H. Zhang, "Commonsense scene graph-based target localization for object search," 2024.

[13] A. A. Ravankar, A. Ravankar, C.-C. Peng, Y. Kobayashi, and T. Emaru, "Task coordination for multiple mobile robots considering semantic and topological information," in *2018 IEEE International Conference on Applied System Invention (ICASI)*, pp. 1088–1091, IEEE, 2018.

[14] A. Chikhalikar, A. A. Ravankar, J. V. S. Luces, S. A. Tafrishi, and Y. Hirata, "An object-oriented navigation strategy for service robots leveraging semantic information," in *2023 IEEE/SICE International Symposium on System Integration (SII)*, pp. 1–6, 2023.

[15] N. Sünderhauf, F. Dayoub, S. McMahon, B. Talbot, R. Schulz, P. Corke, G. Wyeth, B. Upcroft, and M. Milford, "Place categorization and semantic mapping on a mobile robot," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5729–5736, 2016.

[16] R. Martins, D. Bersan, M. Campos, and et al., "Extending maps with semantic and contextual object information for robot navigation: a learning-based framework using visual and depth cues," *Journal of Intelligent and Robotic Systems*, vol. 99, p. 555–569, 2020.

[17] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone, "Kimera: From slam to spatial perception with 3d dynamic scene graphs," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1510–1546, 2021.

[18] A. A. Ravankar, A. Ravankar, T. Emaru, and Y. Kobayashi, "A hybrid topological mapping and navigation method for large area robot mapping," in *2017 56th annual conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 1104–1107, IEEE, 2017.

[19] Y. Wu, Y. Zhang, D. Zhu, Z. Deng, W. Sun, X. Chen, and J. Zhang, "An object slam framework for association, mapping, and high-level tasks," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2912–2932, 2023.

[20] A. Ravankar, A. A. Ravankar, Y. Hoshino, M. Watanabe, and Y. Kobayashi, "Safe mobile robot navigation in human-centered environments using a heat map-based path planner," *Artificial Life and Robotics*, vol. 25, pp. 264–272, 2020.

[21] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[22] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[23] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of field robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[24] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of ECCV 2014: 13th European Conference on Computer Vision, Switzerland*, pp. 740–755, Springer, 2014.

[26] A. Chikhalikar, A. A. Ravankar, J. Victorio Salazar Luces, and Y. Hirata, "Semantic-based multi-object search optimization in service robots using probabilistic and contextual priors," *IEEE Access*, vol. 12, pp. 113151–113164, 2024.

[27] A. Chikhalikar, A. A. Ravankar, J. V. S. Luces, and Y. Hirata, "Integrating semantic awareness and probabilistic priors for object search in indoor environments," in *The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec) 2023*, pp. 1P1–C25, The Japan Society of Mechanical Engineers, 2023.

[28] M. Wang and Liu, "Fuzzy logic based robot path planning in unknown environment," in *2005 International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 813–818 Vol. 2, 2005.

[29] A. A. Ravankar, S. A. Tafrishi, J. V. S. Luces, F. Seto, and Y. Hirata, "Care: Cooperation of ai robot enablers to create a vibrant society," *IEEE Robotics & Automation Magazine*, 2022.

[30] Y. Hirata, J. V. S. Luces, A. A. Ravankar, and S. A. Tafrishi, "Cooperation of assistive robots to improve productivity in the nursing care field," in *The International Symposium of Robotics Research*, pp. 287–294, Springer, 2022.

[31] A. Ravankar, A. Rawankar, and A. A. Ravankar, "Real-time monitoring of elderly people through computer vision," *Artificial Life and Robotics*, vol. 28, no. 3, pp. 496–501, 2023.

[32] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, *et al.*, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[33] V. Furnon and L. Perron, "Or-tools routing library." `https://developers.google.com/optimization/routing/`, May 2024.

[34] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (A. Moschitti, B. Pang, and W. Daelemans, eds.), (Doha, Qatar), pp. 1532–1543, Association for Computational Linguistics, Oct. 2014.