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

# Global Planning for Object Navigation via a Weighted Traveling Repairman Problem Formulation

Ruimeng Liu, Xinhang Xu, Shenghai Yuan*, *Member, IEEE* And Lihua Xie, *Fellow, IEEE*

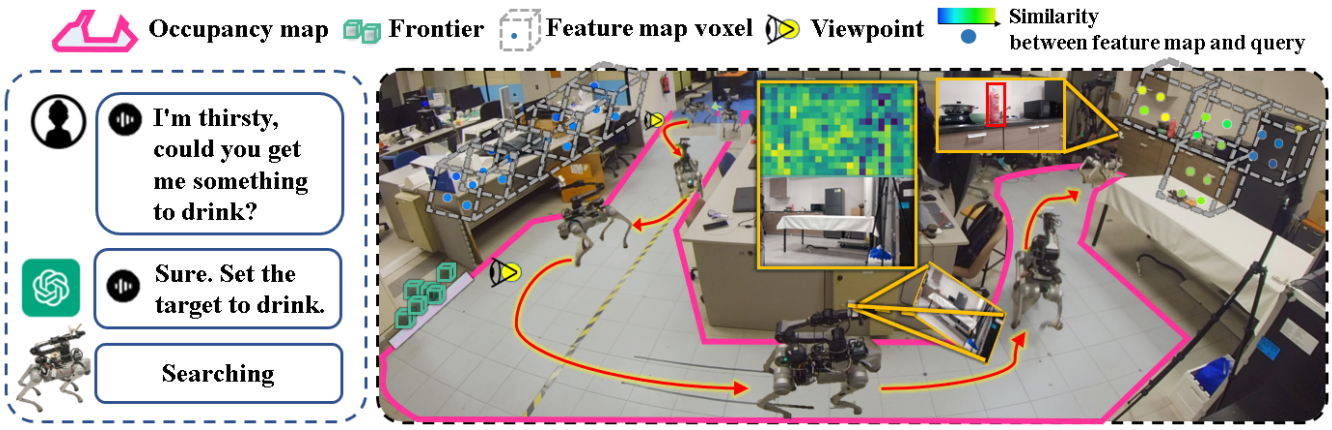


**Fig. 1:** Implementation of our approach in a real robot. A VLM would extract features from an RGB image and compute similarity with the target. Features from the VLM are stored and fused in 3D voxels and would match the embeddings of the query. The right of the figure is a pantry area, so it presents higher similarity than the office area on the left. Frontiers are generated in the margin of free space and unknown space. Viewpoints from frontiers guide the robot to enlarge the explored area.

***Abstract*—Zero-Shot Object Navigation (ZSON) requires agents to navigate to objects specified via open-ended natural language without predefined categories or prior environmental knowledge. While recent methods leverage foundation models or multi-modal maps, they often rely on 2D representations and greedy strategies or require additional training or modules with high computation load, limiting performance in complex environments and real applications. We propose WTRP-Searcher, a novel framework that formulates ZSON as a Weighted Traveling Repairman Problem (WTRP), minimizing the weighted waiting time of viewpoints. Using a Vision-Language Model (VLM), we score viewpoints based on object-description similarity, projected onto a 2D map with depth information. An open-vocabulary detector identifies targets, dynamically updating goals, while a 3D embedding feature map enhances spatial awareness and environmental recall. WTRP-Searcher outperforms existing methods, offering efficient global planning and improved performance in complex ZSON tasks. Code and demos will be available on https://github.com/lrm20011/WTRP_Searcher.**

This work is supported by the National Research Foundation of Singapore under its Medium-Sized Center for Advanced Robotics Technology Innovation. This work was also supported by Chery International under its collaboration projects with Nanyang Technological University. All authors are with the Centre for Advanced Robotics Technology Innovation, School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Email:{ruimeng.liu,xinhang.xu,shyuan,elhxie}@ntu.edu.sg.* Corresponding.

## I. Introduction

Zero-Shot Object Navigation (ZSON) is a challenging task in robotics and AI, where an agent is required to navigate to specified targets using open-ended natural language. Unlike conventional object navigation methods, ZSON operates without a predefined set of object categories or prior knowledge of the environment, making it more adaptable to real-world applications.

Object navigation has received considerable research attention in recent years. Earlier works [1], [2] implicitly utilize the comprehension of the semantic environment through reinforcement or imitation learning but are limited to closed-set settings and certain environments. Recent training-free methods [3]–[5] leverage large language models (LLM) for planning by converting observations into language queries. However, this approach is inefficient and neglects complex 3D correspondence, leading to sub-optimal decisions. Many experiments in biology [6] and robotics [7], [8] have shown that spatial representation and memory are significant mechanisms in human navigation, and explicit maps improve performance and efficiency over end-to-end methods in various navigation tasks. More and more researchers are using prior map-based methods [9]. They build semantic maps [10]–[13] for further inferring or decision making by a closed-set detector or an open-vocabulary Vision-Language Model

(VLM). Some works also combined this with a traditional frontier method, which marks the boundary of observed and unobserved areas for more efficient navigation [14], [15].

Despite these advancements, several challenges remain in map-based and frontier-based approaches. Many methods [10], [12] employ top-down 2D feature maps, which struggle with overlapping objects and occluded environments (e.g., objects inside cabinets). Existing approaches [4], [15] often rely on greedy goal selection strategies, which choose the most promising viewpoint at each step but fail to account for long-term planning. This shortsighted decision-making can lead to inefficient search behaviors in complex environments.

To overcome these challenges, we propose WTRP-Searcher, a novel training-free navigation framework that formulates ZSON as a Weighted Traveling Repairman Problem (WTRP) to optimize search efficiency through global viewpoint selection, as illustrated in Figure 1. Our approach leverages a VLM to compute similarity scores between the environment and object descriptions, projecting them onto a 2D depth-enhanced map for improved spatial reasoning. Frontier-based viewpoint extraction assigns these similarity scores as weights in the WTRP optimization process, enabling more informed goal selection. Additionally, an open-vocabulary object detector dynamically refines candidate viewpoints and navigation targets based on detection confidence, while a 3D embedding feature map continuously updates environment memory, enhancing object localization as the agent explores.

In general, the contributions of this work are as follows:

- We propose WTRP-Searcher, a training-free framework for Zero-Shot Object Navigation (ZSON) that integrates open-vocabulary detection, 3D feature mapping, and a Weighted Traveling Repairman Problem (WTRP) formulation for efficient global planning for object search.
- We introduce a global goal selection policy leveraging items and global associations, addressing the limitations of greedy strategies in prior work.
- We develop a real-time mapping system that builds multi-modal maps, enabling efficient object localization and environment memorization.
- We evaluate our method in simulated and real-world environments, showing superior performance over state-of-the-art approaches. We will open-source our work for the benefit of society.

## II. RELATED WORKS

### A. Frontier-based Navigation

Frontier-based navigation is a widely used approach in exploration tasks, where frontiers mark the boundaries between explored and unexplored areas [16]. Classical methods often employ greedy strategies, selecting the nearest frontier at each step to maximize immediate exploration gains. More advanced strategies, such as next-best-view selection [17], evaluate candidate viewpoints based on information gain and path cost, selecting the one with the highest utility. To improve planning efficiency, recent works [18], [19] have reformulated frontier selection as a Traveling Salesman Problem (TSP), aiming to minimize the total travel cost across all viewpoints, which has demonstrated superior performance compared to traditional heuristic approaches.

Beyond general exploration, frontier-based strategies have been widely adopted in object search problems. Extensions of classical frameworks [20], [21] integrate unknown space exploration with object-centric surface inspection, enabling more target-aware navigation. Several works focus on leveraging frontier selection for object search optimization. For example, [14] introduces a learning-based policy that generates navigation goals based on semantic maps and frontiers, while [22] employs a simpler heuristic, iteratively selecting the closest frontier until an open-vocabulary detector confirms object detection. More recent approaches, such as VLFM [15], combine visual-language similarity scoring with frontier selection, prioritizing viewpoints with the highest confidence to improve search efficiency.

This growing body of research highlights the effectiveness of frontier-based strategies in object search, yet many existing methods still suffer from greedy selection biases, lack of global planning, and reliance on 2D representations, which limit performance in complex, occluded, or long-horizon navigation tasks. Addressing these challenges remains a key direction for improving efficient and scalable object navigation.

### B. Zero-Shot Object Navigation

ZSON requires an agent to navigate toward a specified target without prior training or exposure to the object or environment. Traditional approaches can be broadly categorized into end-to-end learning methods and modular methods. End-to-end methods [23], [24] employ deep reinforcement learning (RL) or supervised learning to directly predict navigation actions from raw sensory inputs. Meanwhile, modular methods [2], [9] construct multi-modal maps that facilitate high-level decision-making and low-level motion planning, improving interpretability and structured reasoning. However, both paradigms struggle with generalization, especially when encountering novel objects and unseen environments.

To address these limitations, recent works have explored foundation models for object navigation, leveraging commonsense knowledge from large-scale internet datasets to improve adaptability. Some approaches [3]–[5] incorporate LLMs to generate text-based object queries or reason about object locations, enabling training-free exploration. Other methods [11], [25], [26] integrate VLMs as additional input to learning-based frameworks, enhancing scene understanding and target recognition. Additionally, recent works [12], [15] fuse VLMs with modular exploration methods, using semantic similarity scores to rank and prioritize navigation goals.

While these advances significantly improve zero-shot navigation capabilities, several challenges remain, including long-term spatial reasoning, efficient goal selection, and real-

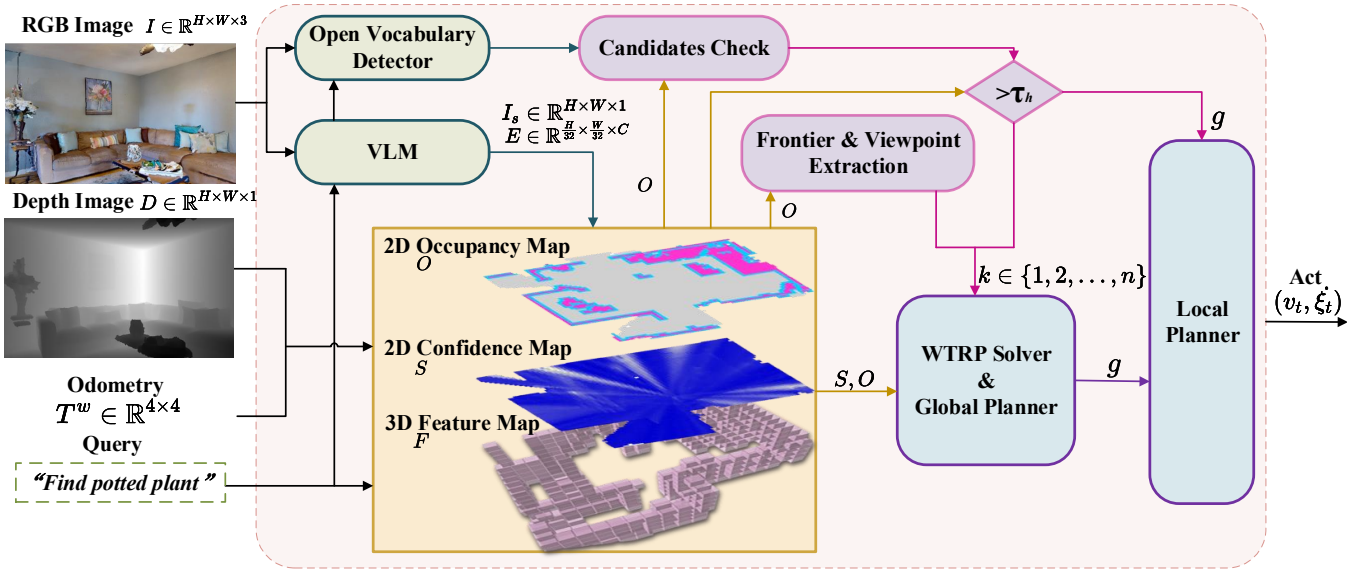


**Fig. 2:** Overview of our framework. Sub-modules are colored and classified according to their functionality. VLM and open-vocabulary detector percept the environment and output features for mapping and planning; The mapping module constructs maps of three types from features, depth images, and poses; Modules in purple leverage detection results and multi-modal maps to generate candidate points for mid-term goals and potential navigation goals; All the candidate points are in overall planning by WTRP solver to get a global optimum route and the path to the first node would be sent to local planner to generate actions

world deployment feasibility. Addressing these limitations is crucial for enabling robust, generalizable, and scalable ZSON solutions.

## III. METHODOLOGY

In this section, we present the system architecture and the proposed methods in detail, outlining how our approach integrates global planning, semantic understanding, and multi-modal mapping to achieve efficient zero-shot object navigation.

### A. Problem Formulation and System Overview

In a ZSON task, the agent must explore an unfamiliar environment to locate objects that match open-ended natural language descriptions. Unlike conventional navigation approaches that rely on predefined object categories or prior scene knowledge, zero-shot navigation requires the agent to interpret semantic queries dynamically and infer target locations in real time.

To achieve this, the agent receives an RGB image stream, an aligned depth image, and pose information as inputs. These sensory data are used to construct a spatial representation of the environment, allowing the agent to reason about potential object locations. Based on this representation, the agent plans a trajectory or generates a sequence of actions to navigate efficiently toward the target object. The overall architecture of our framework is illustrated in Figure 2, detailing the integration of multi-modal perception, mapping, and goal selection to enable efficient object search.

### B. Map Construction

Given an RGB image $I \in \mathbb{R}^{H\times W\times 3}$, we extract features and project points into world space using an aligned depth image $D \in \mathbb{R}^{H\times W\times 1}$, camera parameters, and the pose transformation matrix $T^w \in \mathbb{R}^{4\times 4}$. This process constructs three complementary maps: a 2D occupancy grid map $O$ for navigation and exploration, a 2D semantic confidence map $S$ for goal selection, and a 3D feature map $F$ for object matching.

*1) Occupancy Grid Map:* Each cell in the 2D occupancy grid map holds a probability value and an inflation count, indicating whether it is occupied, free, unknown, or influenced by nearby occupied grids. The map is generated by projecting depth points onto a 2D grid, applying raycasting to classify grid cells as hits or misses, and updating their state using a Bayesian approach. After each update, surrounding grids are adjusted to reflect obstacle inflation, improving navigation safety.

*2) Semantic Confidence Map:* To integrate semantic awareness into navigation, we construct a 2D semantic confidence map using vision-language embeddings. Inspired by [12], given an RGB image $I \in \mathbb{R}^{H\times W\times 3}$, we use a CLIP-based feature extractor [27], [28] to obtain patch-level embeddings $E \in \mathbb{R}^{H/32\times W/32\times C}$, where $C$ denotes the embedding dimension. The cosine similarity between target descriptions and image patches is computed and then upsampled to the resolution of the input image, producing a per-pixel similarity representation, $I_s \in \mathbb{R}^{H\times W\times 1}$.

These 2D similarity values are then projected into 3D world space using the aligned depth image $D \in \mathbb{R}^{H\times W\times 1}$ and the pose transformation matrix $T^w \in \mathbb{R}^{4\times 4}$. To enhance robustness, the observed similarity score of each grid cell $p$ in time $t$, denoted as $S_p^t$, is computed as the average of the top 20% similarity values among all points projected into that cell in this observation. The covariance of the similarity value at grid cell $p$ is then computed as:

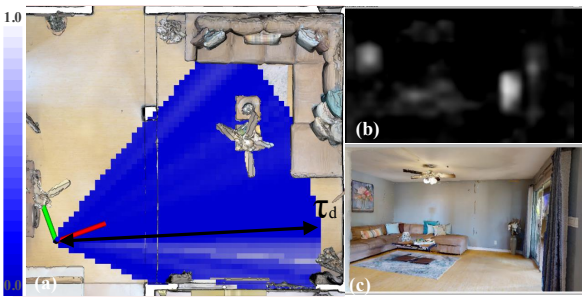


**Fig. 3:** Construct a temporal semantic confidence map. (a) is the construction result. (b) is a pixel-wise confidence image generated from (c) by VLM. The mapping is limited in $\tau_d$ while points outside the region would also slightly contribute to the updating to enhance semantic guidance for navigation.

$$\sigma_p^{2\,t} = \frac{\alpha_c * (d_{p,p_0})^2}{n_p}, \tag{1}$$

where $n_p$ is the number of points in the 2D grid, $d_{p,p_0}$ represents the distance of the grid to the sensor origin $p_0$, and $\alpha_c$ is a scaling factor.

To maintain consistency over time, we employ a Kalman filter to update the similarity value $S_p$ in grid $p$ using the current observation $S_p^t$:

$$K = \frac{\sigma_p^{2\,t}}{\sigma_p^{2-} + \sigma_p^{2\,t}}, \tag{2}$$

$$S_p = S_p^t + K(S_p^- - S_p^t), \tag{3}$$

$$\sigma_p^2 = (1-K)\sigma_p^{2\,t}. \tag{4}$$

To evaluate viewpoints in free space, similarity values for these grids $p'$ are estimated by raycasting from the sensor to the grids hit by depth points $p$ :

$$S_{p'}^t = S_p^t - \gamma d_{p,p'}, \tag{5}$$

$$\sigma_{p'}^{2\,t} = \beta \sigma_p^{2\,t}, \tag{6}$$

where $\gamma$ and $\beta$ control similarity decay over distance.

Additionally, to mitigate depth sensor noise, the updates are restricted to regions within a distance threshold $\tau_d$ from the sensor. Figure 3 illustrates the process of constructing a temporal confidence map $S^t$ from an observation.

The confidence map is defined in a task-specific manner. For a newly specified target, the previous confidence map is reset. When multiple targets are searched simultaneously, an individual confidence map is updated and maintained for each target. The final confidence at each location is then obtained as the maximum across all confidence maps.

*3) 3D Feature Map:* While the 2D confidence map is typically constructed for a specific task, maintaining a unified semantic memory of the environment would allow new tasks to leverage previously acquired knowledge, reducing search time and improving efficiency. Following the procedure described in subsubsection III-B2, at each time step $t$, we project feature embeddings into 3D space to form $F^t$ using aligned depth images and pose information. The average embedding $F_p^t \in F^t$ is computed for all embeddings mapped to the same 3D grid cell $p$. To maintain temporal consistency, we apply a Kalman filter to fuse $F_p^t$ with the previous feature map $F_p$, with the covariance and update equations following Equation 1 and Equation 2, respectively. For efficient storage and traversal, we employ a modified octree structure [29], which exclusively stores occupied grids with embeddings. Additionally, grids corresponding to the floor and ceiling are systematically eliminated based on height thresholds, further optimizing the representation.

Since the VLM we employ produces patch-level features $E \in \mathbb{R}^{H/32\times W/32\times C}$, bilinear interpolation is typically used to obtain image-level features $E' \in \mathbb{R}^{H\times W\times C}$. Directly upsampling patch level features $E$ to pixel level features $E'$, projecting it with aligned depth points from $D \in \mathbb{R}^{H\times W\times 1}$ to 3D grids, and then fusing features in the same voxel can be a straightforward method.

$$\mathcal{R}(x,y) = \left\{ (u,v) \in \mathbb{Z}^2 : \begin{array}{c} u \in \left\{ \left\lfloor \frac{x}{r} \right\rfloor, \left\lfloor \frac{x}{r} \right\rfloor + 1 \right\}, \\ v \in \left\{ \left\lfloor \frac{y}{r} \right\rfloor, \left\lfloor \frac{y}{r} \right\rfloor + 1 \right\} \end{array} \right\}. \tag{7}$$

$$E'(x,y) = \sum_{(u,v)\in\mathcal{R}(x,y)} E(u,v) \cdot w_{u,v}(x,y), \tag{8}$$

where the bilinear weights are

$$w_{u,v}(x,y) = \begin{cases} \left(1 - \left|\frac{x}{r} - u\right|\right)\left(1 - \left|\frac{y}{r} - v\right|\right), & (u,v) \in \mathcal{R}(x,y), \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Here $r$ is the upsample factor, which is 32 in the scenario. $u, v$ are the related indices in $E$ of pixel $x, y$ in $E'$.

However, when the dimension $C$ is large, performing interpolation and fusion introduces significant computational and storage overhead. On the contrary, we integrate the prior-interpolation step with the fusion step internally and use the summarization of bilinear weights to replace the feature fusion in bilinear interpolation: when projecting the aligned depth points to 3D voxels, in each voxel $p$, we would count its related pixels in $E$ expressed as $\mathcal{D}_p$ and the corresponding weight.

$$\mathcal{W}_{u,v}^p = \sum_{(x,y)\in\mathcal{D}_p} w_{u,v}(x,y) \tag{10}$$

After the projection, we compute the current feature projected to voxel $p$ with the weighted average method.

$$F_p^t = Norm\left(\frac{1}{\sum_{(u,v)\in\mathcal{R}(\mathcal{D}_p)} \mathcal{W}_{u,v}^p} \sum_{(u,v)\in\mathcal{R}(\mathcal{D}_p)} \mathcal{W}_{u,v}^p E(u,v)\right) \tag{11}$$

Because points projected to the same voxel are near, we simply assume they share the same covariance. When fusing $F_p^t$ with the previous observation $F_p$ following Equation 1 and Equation 2, the $d_{p,p_0}$ is computed as the center of the voxel $p$ to the sensor. This strategy significantly reduces the number of high-dimensional embeddings processed during fusion.

### C. Candidate Generation Policy

Candidate points for mid-term goals in object search are selected from frontier cluster viewpoints, detections from an open-vocabulary object detector, and feature map matching results.

In an unknown environment, frontiers are extracted from free-space grid cells adjacent to unknown regions with the occupancy map. Viewpoints are then generated by clustering these frontiers in previously visited free space, ensuring an optimal vantage point for exploration.

During navigation, an open-vocabulary detector continuously searches for target categories. We utilize YOLO-World [30] for real-time object detection. If the confidence score of a detected object exceeds the threshold $\tau_h$, its corresponding viewpoint is immediately set as the next navigation goal $g$. For objects with confidence scores between $\tau_l$ and $\tau_h$, the viewpoint is added as a candidate point for consideration in the WTRP solver. Since high-confidence detections can sometimes be false positives due to specific viewing angles or distances, detected targets are re-evaluated during navigation. If the confidence score drops below $\tau_l$ for a sustained period, the target is removed from the candidate list.

As the agent navigates, the 3D feature map is built and continuously updated. This map enhances the system's environment memorization capability, improving target localization. Given a language-based query, we compute the similarity between the description and stored embeddings in each 3D grid. Similar to object detections, locations corresponding to high-confidence grid cells are directly treated as navigation goals $g$, while those with slightly lower confidence are incorporated into the WTRP solver for optimized selection.

### D. Mid-term Goal Choosing Policy

This section describes the method to choose the mid-term goal when candidates and related information are given. Recent advancements in exploration tasks have formulated the problem of visiting all viewpoints as a Traveling Salesman Problem (TSP), aiming to minimize the length of the global tour. However, directly applying this formulation to object search tasks may not be optimal, as the agent stops once it finds a target—meaning that, in most cases, not all viewpoints are visited. Additionally, from a behavioral perspective, viewpoints with higher semantic similarity to the target should be prioritized over purely minimizing the travel distance. A global tour optimization that neglects semantic relevance may result in suboptimal search efficiency.

Following [31], we define the Weighted Traveling Repairman Problem (WTRP) as follows:

Given a set of points $k \in \{0, 1, ..., n\}$ with associated weights $W_k$, a cost matrix $M_{k_r,k_s}$ representing the time cost between any two points $k_r$ and $k_s$, and a start point $0 \in \{0, 1, ..., n\}$, the goal is to find a permutation $\pi_0 = 0, \pi_1, ..., \pi_n$ that minimizes:

$$\min_{\pi} \sum_{i=1}^{n} W_{\pi_i} \sum_{j=1}^{i} M_{\pi_{j-1},\pi_j}, \tag{12}$$

where $\pi_j$ denotes the $j$-th node to be visited, $\pi_0$ is the agent's current position, and $W_{\pi_j}$ is the weight of the node, computed as:

$W_{\pi_j}$ denotes the weight of nodes and can be gained from:

$$W_{\pi_j} = \phi(S_{p_{\pi_j}}), \tag{13}$$

$$\phi(x) = \alpha_s e^{\mu x+\nu}, \tag{14}$$

where $\phi(x)$ is a mapping function designed to amplify the differences in node weights. The parameters $\alpha_s$, $\mu$, and $\nu$ are scaling factors.

Here, $p_k$ represents the position of a candidate or the agent in the map, and $S_{p_{\pi_j}}$ is the similarity of the node. If the candidate is the viewpoint from clusters, this can be retrieved from the 2D semantic confidence map based on its location. If the node is from the detection result of an open-vocabulary detector or the match result from the 3D feature map, $S_{p_{\pi_j}}$ is the detection confidence or match similarity, respectively.

Inspired by [18], [32], we incorporate both the distance between nodes and the agent's motion state—where $v_0$ and $\xi_0$ denote the agent's current velocity and orientation, while $v_{max}$ and $\dot{\xi}_{max}$ represent its maximum linear and angular velocities. $\xi_k$ describes the orientation of each candidate node. The cost matrix is then defined as:

$$t(k_r, k_s) = max\{\frac{d_{p_{k_r},p_{k_s}}}{v_{max}}, \frac{|\xi_{k_r} - \xi_{k_s}|}{\dot{\xi}_{max}}\}, \tag{15}$$

$$c_c(k) = cos^{-1}\frac{(p_k - p_0)\cdot v_0}{\|p_k - p_0\|\|v_0\|}, \tag{16}$$

$$c_s(k) = \frac{h_k}{h_{max}}, \tag{17}$$

$$M_{0,k} = t(0, k) + w_c \cdot c_c(k) + w_f \cdot c_s(k), \quad k \in 1, 2, ..., n, \tag{18}$$

$$M_{k_r,k_s} = M_{k_s,k_r} = t(k_r, k_s), \quad k_r, k_s \in 1, 2, ..., n, \tag{19}$$

$$M_{k,0} = 0, \tag{20}$$

where $t(k_r, k_s)$ denotes the travel time between nodes $k_r$ and $k_s$, while $c_c(k)$ ensures motion consistency. The term $c_s(k)$ is introduced to encourage the agent to first explore small, enclosed, unexplored areas, which often result from incomplete observations and could otherwise cause inefficient back-and-forth maneuvers. This item is only used for candidates of viewpoints from clusters.

To compute $h_k$, a ray is cast from the viewpoint toward the cluster center of its frontiers. If the ray intersects observed grids within the distance threshold $h_{max}$, $h_k$ is set as the distance from the hit grid to the cluster center. Otherwise, $h_k$ defaults to $h_{max}$.

We solve the WTRP using the LKH solver [33]. The first node in the computed tour serves as the mid-term goal and would be sent to the local planner subsection III-E. If the agent reaches the goal with candidates generated in the tour, or if the execution time exceeds the replan

**TABLE I:** Comparison with baselines in single-object navigation.

| Method | Training-free | Foundation Model | HM3D | |
|---|---|---|---|---|
| | | | SR↑(%) | SPL↑(%) |
| ZSON [25] | × | CLIP | 25.5 | 12.6 |
| Voronav [36] | ✓ | GPT-3.5 | 42.0 | 26.0 |
| ESC [4] | ✓ | GLIP, GPT-3.5 | 39.2 | 22.3 |
| VLFM [15] | × | BLIP2 | 52.5 | 30.4 |
| OneMap [12] | ✓ | CLIP | 55.8 | 37.4 |
| Ours | ✓ | CLIP | **59.2** | **39.8** |

threshold $\tau_t$, the global tour is recomputed to ensure up-to-date planning; otherwise, the agent traverses the nodes of the tour in sequence.

### *E. Local Planner*

Once a navigation goal $g$ is selected, a suitable path is generated based on the distance from the agent's current position. If the target is farther away, waypoints are computed using the A* algorithm, whereas single-point goals are directly sent to the local planner. For trajectory optimization, we adapt the local planner from [34], leveraging MINCO [35] to ensure smooth motion planning. The final linear and angular velocity commands $(v_t, \dot{\xi}_t)$ for the chassis are generated by a low-level controller. The local planner is capable of handling dynamic obstacles and producing smooth, collision-free trajectories, thereby enhancing the system's robustness and overall performance.

## IV. EXPERIMENTS

In this section, we evaluate our method using the Habitat simulator on the HM3D dataset [37] by comparing it with baseline methods in both single-object and multi-object navigation tasks. We further validate our approach through real-world experiments.

### *A. Single Object Navigation*

This task requires the agent to locate an object belonging to the target category in an unknown environment. We evaluate our method using the validation split of the HM3D dataset, which consists of 2000 single-object navigation episodes across 20 scenes and 6 object categories.

**Evaluation Metrics:** We use standard evaluation metrics, including Success Rate (SR) in % and Success weighted by Path Length (SPL) in %. SR measures the proportion of successful episodes, while SPL accounts for both success and path efficiency, computed as the ratio of the optimal path length to the actual path length, weighted by a binary success indicator.

**Baseline Selection:** We compare our approach against the current state-of-the-art (SOTA) methods in zero-shot object navigation, including ZSON [25], VoroNav [36], ESC [4], VLFM [15] and Onemap [12]. ZSON is a learning-based method that incorporates VLM embeddings as input. VoroNav constructs a Voronoi graph from a semantic map, enabling a large language model (LLM) to reason about waypoints. ESC and VLFM both employ frontiers for navigation:


**Fig. 4:** Comparison of 3D and 2D feature map. (a),(b) is the result of 3D and 2D feature map, respectively; the colored spheres indicate the likelihood of target presence, where red denotes higher confidence regions. (c) is the scene, where targets are marked with a white circle in the figure.

**TABLE II:** Multi-object navigation evaluation

| Method | Type | HM3D | | | |
|---|---|---|---|---|---|
| | | SR↑(%) | SPL↑(%) | PR↑(%) | PPL↑(%) |
| TSP Exp | Simultaneous | 29.3 | 12.1 | 49.3 | 17.8 |
| W/o WTRP | | 52.1 | 17.1 | 67.8 | 22.4 |
| Ours | | **53.4** | **23.0** | **68.9** | **28.9** |
| OneMap [12] | Sequential | 54.2 | 27.8 | 65.5 | 32.7 |
| Ours | | **57.2** | **33.4** | **67.1** | **38.2** |

ESC utilizes a VLM for scene understanding and an LLM for goal selection, while VLFM ranks frontiers based on VLM similarity scores, selecting the highest-scoring frontier as the next waypoint. OneMap uses patch-level CLIP and constructs a 2D feature map storing VLM embeddings.

### *B. Multi-Object Navigation*

For multi-object navigation tasks, we design two types of experiments. In the first setting, the agent is given a list of target categories at the beginning and is required to find at least one object for each category. In the second setting, the agent receives the target categories sequentially, where a new target is assigned only after the current one is found. We evaluate our approach on the benchmark from [12], which consists of 236 episodes in 20 scenes across six categories. Each episode includes three target categories, and the agent has no prior knowledge of the scene.

**Evaluation Metrics:** In addition to the metrics defined in subsection IV-A, we introduce additional metrics for multi-object navigation: Progress Rate (PR) in %, which measures the rate at which the agent makes progress toward finding all target objects, and Progress Weighted by Path Length (PPL) in %, defined as:

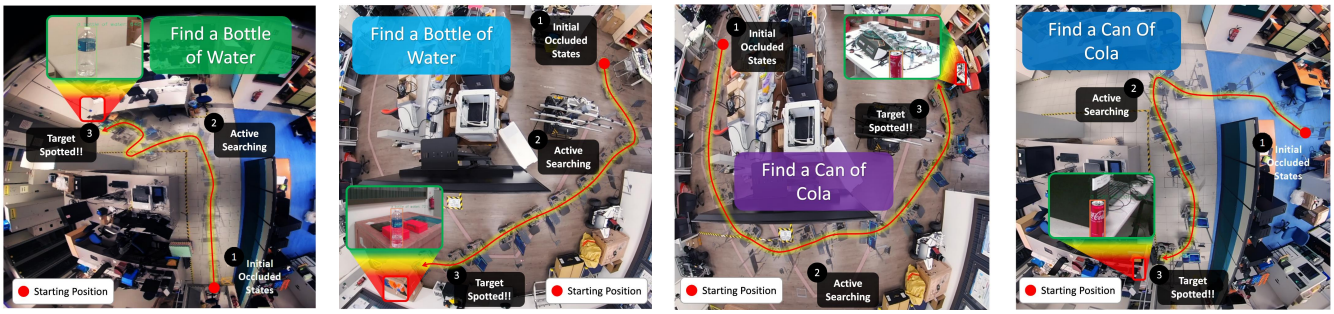


**Fig. 5:** The process of visual language navigation in two real office scenes over two categories.

$$\text{PPL} = \frac{1}{N}\sum_{i=1}^{N} r_i \frac{l_i}{\max(p_i, l_i)},$$

where $N$ is the number of episodes, $l_i$ is the optimal path length for episode $i$, and $r_i$ and $p_i$ represent the PR and the actual total path length of episode $i$, respectively.

**Baseline Selection:** We compare our method with OneMap [12], which is designed for sequential multi-object navigation. As no recent open-source approaches are available for simultaneous multi-object navigation, we conduct an ablation study to evaluate the impact of our framework's key components. Specifically, it includes replacing the WTRP solver with a greedy strategy ("W/o WTRP"), and adopting a TSP-based exploration framework that explores and detects the environment separately ("TSP Exp"), as shown in Table II.

### C. Results Discussion

The results of our comparison are presented in Table I and Table II. Our method achieves superior performance over all zero-shot SOTA baselines in both SR and SPL.

Specifically, we achieve SRs of 59.2% and 56.8% in single- and multi-object navigation, corresponding to SPL improvements of 6.1% and 5.8% over OneMap. We attribute our higher SR to our candidate selection policy, where rechecking detected objects reduces the false positive rate, leading to more reliable target identification. Additionally, the use of patch-level CLIP embeddings (used by ours and OneMap) enables a more detailed and structured understanding of the environment than image-level visual-language models (used by VLFM), allowing the constructed semantic confidence and feature map to guide exploration more effectively and precisely.

We achieve a 6.3% SPL improvement in single-object navigation, and improvements of 20% in SPL and 16.8% in PPL in multi-object navigation over OneMap. These improvements indicate that formulating the route planning problem as a WTRP leads to more efficient trajectories compared to frontier-based approaches that greedily select the highest-scoring frontier. The ablation study in simultaneous multi-object navigation evaluation also supports that.

### D. Qualitative Analysis

In environments with stacked spatial layouts, 3D feature maps exhibit superior performance in small object localization. For instance, as shown in Figure 4, in a kitchen with cabinets above and a stove below, the 3D feature map more accurately highlights the spatial region containing the target object, such as a kettle. The 3D feature map's capability to store and match environmental embeddings enables the agent to efficiently revisit high-similarity regions, reducing redundant exploration.

**TABLE III:** Time consumption of sub-modules on ORIN. In a Desktop PC, the performance can be doubled. We show the worst-case performance. An asterisk (*) indicates that the sub-module may also be triggered by external signals.

| Task | Time consumption (s) | Desired interval (s) |
|---|---|---|
| VLM | 0.091 | 0.125 |
| Open-vocabulary detector | 0.079 | 0.125 |
| 2D grid map | 0.108 | 0.125 |
| 2D confidence map | 0.129 | 0.125 |
| 3D feature map | 0.589 | 0.250 |
| WTRP solver | 0.010 | 8.000* |
| Trajectory optimization | 0.005 | 0.500* |

### E. Real World Experiments

We deploy our method on a differential-drive robot equipped with a RealSense D455 depth camera, which provides RGB-D images. For odometry estimation, we utilize RTAB-Map [38], leveraging both RGB-D images and IMU data from the D455 camera. The entire system is implemented on a Jetson Orin NX 16GB platform.

The time consumption of each sub-module per iteration, along with its desired updating interval, is summarized in Table III. These results demonstrate the efficiency of our system, highlighting its feasibility for real-world robotic deployment. Although the 3D feature map looks a bit heavy for the embedded device, the module serves to match language with the environment when receiving queries, and the latency in construction does not influence search. Figure 5 illustrates top-down trajectories across four distinct scenarios, demonstrating the agent's ability to navigate around objects effectively. Further implementation details can be found in the accompanying video.

## V. CONCLUSION

In this work, we present WTRP-Searcher, a framework for ZSON that casts the task as a Weighted Traveling Repairman Problem (WTRP). The method combines frontier-based exploration, open-vocabulary detection, and 3D feature

matching to select candidate viewpoints, with WTRP guiding global planning and a local planner ensuring obstacle-aware execution. Experiments show that WTRP-Searcher surpasses state-of-the-art baselines in efficiency, accuracy, and adaptability across both simulation and real-world tests. These results highlight the value of uniting spatial reasoning, semantic cues, and global optimization for ZSON. Future work will investigate lifelong and domain adaptation to strengthen robustness and sim-to-real generalization.

## References


[1] E. Wijmans, A. Kadian, A. 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 (ICLR)*, 2020.

[2] M. Chang, A. Gupta, and S. Gupta, "Semantic visual navigation by watching youtube videos," *Advances in Neural Information Processing Systems*, 2020.

[3] B. Yu, H. Kasaei, and M. Cao, "L3mvn: Leveraging large language models for visual target navigation," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[4] K. Zhou, K. Zheng, C. Pryor, Y. Shen, H. Jin, L. Getoor, and X. E. Wang, "Esc: Exploration with soft commonsense constraints for zero-shot object navigation," in *International Conference on Machine Learning*, 2023.

[5] V. S. Dorbala, J. F. Mullen, and D. Manocha, "Can an embodied agent find your "cat-shaped mug"? llm-based zero-shot object navigation," *IEEE Rob. Autom. Lett.*, 2024.

[6] R. G. Golledge, *Wayfinding behavior: Cognitive mapping and other spatial processes*. JHU press, 1999.

[7] D. S. Chaplot, R. Salakhutdinov, A. Gupta, and S. Gupta, "Neural topological slam for visual navigation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[8] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," in *International Conference on Learning Representations (ICLR)*, 2020.

[9] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," *Advances in Neural Information Processing Systems*, 2020.

[10] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2023, pp. 10 608–10 615.

[11] M. Wei, T. Wang, Y. Chen, H. Wang, J. Pang, and X. Liu, "Ovexp: Open vocabulary exploration for object-oriented navigation," *arXiv preprint arXiv:2407.09016*, 2024.

[12] F. Lukas Busch, T. Homberger, J. Ortega-Peimbert, Q. Yang, and O. Andersson, "One map to find them all: Real-time open-vocabulary mapping for zero-shot multi-object navigation," *arXiv e-prints*, pp. arXiv–2409, 2024.

[13] B. Yu, Y. Liu, L. Han, H. Kasaei, T. Li, and M. Cao, "Vln-game: Vision-language equilibrium search for zero-shot semantic navigation," *arXiv preprint arXiv:2411.11609*, 2024.

[14] B. Yu, H. Kasaei, and M. Cao, "Frontier semantic exploration for visual target navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023.

[15] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "Vlfm: Vision-language frontier maps for zero-shot semantic navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2024.

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

[17] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon path planning for 3d exploration and surface inspection," *Autonomous Robots*, 2018.

[18] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning," *IEEE Rob. Autom. Lett.*, 2021.

[19] Y. Zhao, L. Yan, H. Xie, J. Dai, and P. Wei, "Autonomous exploration method for fast unknown environment mapping by using uav equipped with limited fov sensor," *IEEE Trans. Ind. Electron.*, 2023.

[20] S. Papatheodorou, N. Funk, D. Tzoumanikas, C. Choi, B. Xu, and S. Leutenegger, "Finding things in the unknown: Semantic object-centric exploration with an mav," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2023.

[21] Y. Luo, Z. Zhuang, N. Pan, C. Feng, S. Shen, F. Gao, H. Cheng, and B. Zhou, "Star-searcher: A complete and efficient aerial system for autonomous target search in complex unknown environments," *IEEE Rob. Autom. Lett.*, 2024.

[22] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[23] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, D. Kumaran, and R. Hadsell, "Learning to navigate in complex environments," in *International Conference on Learning Representations (ICLR)*, 2017.

[24] R. Ramrakhya, E. Undersander, D. Batra, and A. Das, "Habitat-web: Learning embodied object-search strategies from human demonstrations at scale," in *Proc. of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

[25] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra, "Zson: Zero-shot object-goal navigation using multimodal goal embeddings," *Advances in Neural Information Processing Systems*, 2022.

[26] Q. Zhao, L. Zhang, B. He, H. Qiao, and Z. Liu, "Zero-shot object goal visual navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2023.

[27] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[28] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021.

[29] D. Duberg and P. Jensfelt, "Ufomap: An efficient probabilistic 3d mapping framework that embraces the unknown," *IEEE Rob. Autom. Lett.*, 2020.

[30] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[31] F. Afrati, S. Cosmadakis, C. H. Papadimitriou, G. Papageorgiou, and N. Papakostantinou, "The complexity of the travelling repairman problem," *RAIRO-Theoretical Informatics and Applications*, 1986.

[32] Y. Zhao, L. Yan, H. Xie, J. Dai, and P. Wei, "Autonomous exploration method for fast unknown environment mapping by using uav equipped with limited fov sensor," *IEEE Trans. Ind. Electron.*, 2023.

[33] R. Tinós, K. Helsgaun, and D. Whitley, "Efficient recombination in the lin-kernighan-helsgaun traveling salesman heuristic," in *Proc. 15th Int. Conf. Parallel Problem Solving from Nature (PPSN XV)*, 2018.

[34] X. Zhou, X. Wen, Z. Wang, Y. Gao, H. Li, Q. Wang, T. Yang, H. Lu, Y. Cao, C. Xu *et al.*, "Swarm of micro flying robots in the wild," *Science Robotics*, 2022.

[35] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE Transactions on Robotics*, 2022.

[36] P. Wu, Y. Mu, B. Wu, Y. Hou, J. Ma, S. Zhang, and C. Liu, "Voronav: Voronoi-based zero-shot object navigation with large language model," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

[37] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang *et al.*, "Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai," *arXiv preprint arXiv:2109.08238*, 2021.

[38] 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*, 2019.