ApexNAV: An Adaptive Exploration Strategy for Zero-Shot Object Navigation With Target-Centric Semantic Fusion

Mingjie Zhang , Yuheng Du, Chengkai Wu , Jinni Zhou, Zhenchao Qi, Jun Ma , *Senior Member, IEEE*, and Boyu Zhou , *Member, IEEE*

Abstract—Navigating unknown environments to find a target object is a significant challenge. While semantic information is crucial for navigation, relying solely on it for decision-making may not always be efficient, especially in environments with weak semantic cues. Additionally, many methods are susceptible to misdetections, especially in environments with visually similar objects. To address these limitations, we propose ApexNav, a zero-shot object navigation framework that is both more efficient and reliable. For efficiency, ApexNav adaptively utilizes semantic information by analyzing its distribution in the environment, guiding exploration through semantic reasoning when cues are strong, and switching to geometry-based exploration when they are weak. For reliability, we propose a target-centric semantic fusion method that preserves long-term memory of the target and similar objects, enabling robust object identification even under noisy detections. We evaluate ApexNav on the HM3Dv1, HM3Dv2, and MP3D datasets, where it outperforms state-of-the-art methods in both SR and SPL metrics. Comprehensive ablation studies further demonstrate the effectiveness of each module. Furthermore, real-world experiments validate the practicality of ApexNav in physical environments.

Index Terms—Search and rescue robots, vision-based navigation, autonomous agents.

I. INTRODUCTION

OBJECT goal navigation (ObjectNav) requires an autonomous agent to navigate to a target object specified by its category in an unknown environment [2], which is a fundamental challenge in Robotics and AI. With rapid advancements in Large Language Models (LLMs) and Vision-Language Models (VLMs), which encode rich commonsense knowledge, zero-shot ObjectNav (ZSON) approaches have emerged [6], [7], [9], [14], [26], [27], [28], [30], [32]. These methods significantly advance the field by offering superior generalization and sim-to-real transfer capabilities compared to data-driven approaches [5], [19], [25].

Nonetheless, current ZSON methods still face challenges in efficiency and robustness. First, most methods [26], [27], [28], [30] rely heavily on semantic reasoning to explore and search. However, semantic guidance is not always available or reliable. For example, at the beginning of task, the agent may face a blank wall or occlusions, limiting semantic cues and hindering effective semantic-based navigation. Moreover, ambiguous or weak semantic cues can mislead the agent. For example, a plant might appear in various rooms like a bedroom, living room, or bathroom, and on different surfaces like a table, windowsill, or floor, making it hard to infer its location without sufficient context. In summary, over-reliance on semantics can reduce navigation efficiency. Secondly, some approaches [7], [27] rely on single-frame detections to identify targets, while others [9], [28] adopt max-confidence fusion to build semantic maps for target identification. However, relying solely on single-frame results often lacks robustness. The max-confidence fusion approach also struggles with misdetections, as even sporadic high-confidence misdetections can result in incorrect semantic labels that are hard to correct through subsequent observations. These issues are exacerbated in cluttered scenes with occlusions or similar-looking objects. Therefore, improving robustness under noisy detection requires multi-frame integration and more reliable fusion strategies.

In the letter, we address these limitations by drawing inspiration from human behavior in searching objects. Humans tend to quickly explore their surroundings when semantic cues are weak and concentrate their search in likely target regions when cues are strong. Furthermore, when encountering a potentially target-like object, humans do not make immediate decisions under uncertainty. Instead, they gather additional observations to accumulate evidence and refine their judgment. Building on these insights, we propose **ApexNav**, an efficient and reliable framework for ZSON, with two key contributions:

1) *Adaptive exploration strategy*: ApexNav makes more effective use of semantic information by analyzing the environment's semantic distribution. When strong semantic cues are available, it leverages semantic reasoning to guide the agent toward likely target regions. In contrast, when semantic information is

limited, it switches to a geometry-based mode to rapidly explore unknown areas. To further improve efficiency, semantic-mode frontier selection is formulated as a Traveling Salesman Problem, optimizing the visiting order of high-scoring frontiers and avoiding greedy selection [26], [27].

2) *Target-centric semantic fusion*: ApexNav maintains long-term memory of the target object and visually similar objects. By aggregating multi-frame observations with context-aware confidence weighting, it enables reliable target identification even when using noisy and imperfect detectors.

We evaluate ApexNav on the HM3Dv1 [18], HM3Dv2 [24], and MP3D [4] datasets, where it achieves state-of-the-art zero-shot performance. Notably, it achieves relative improvements in SR and SPL of 5.5% and 8.6% on HM3Dv1, and 19.8% and 16.9% on HM3Dv2 over previous methods, establishing a new baseline for ZSON. We further conduct ablation studies to validate the contribution of each module and analyze failure cases to provide insights for future work. Extensive real-world experiments also confirm the practicality of ApexNav.

II. RELATED WORK

A. Object Goal Navigation

ObjectNav methods are generally divided into end-to-end and modular approaches. End-to-end methods [25], based on reinforcement learning require heavy training and have high computational costs. Afterwards, modular learning-based methods [5], [19] reduce training effort but do not fully solve the problem. Recent studies introduce LLMs [1], [11], [20] and VLMs [10], [17] into ObjectNav, enabling zero-shot methods [6], [7], [9], [26], [27], [28], [30], [32] that achieve strong performance without additional training. Building on this, we further improve efficiency and reliability through an adaptive exploration strategy and a target-centric semantic fusion mechanism.

B. Exploration Strategy for ZSON

Current ZSON methods mainly use frontier-based exploration. CoW [7] follows a nearest-frontier strategy based only on geometry, ignoring semantic cues in indoor environments. L3MVN [28] and TriHelper [30] construct a semantic map and use LLMs to identify target-related frontiers, while SG-Nav [26] leverages 3D scene graphs to prompt LLMs for reasoning. VLFM [27] applies a VLM to build a value map linking frontiers to the target. These methods typically score frontiers with LLMs or VLMs and use a greedy policy to select the highest-scoring one. WTRP-Searcher [12] instead models ObjectNav as a Weighted Traveling Repairman Problem to overcome greedy policy limitations. While effective in semantically rich environments, they often perform poorly when semantic cues are weak. SemUtil [6] notes that early semantic guidance is inefficient and adopts a fixed two-stage strategy that starts with geometry for 50 steps and then switches to semantics. However, this fixed scheme lacks adaptability. Moreover, SemUtil ranks frontiers using a utility function (semantic score divided by path length), the trade-off between semantic relevance and distance may suppress promising frontiers and limit overall performance. To address these, we propose an adaptive strategy that effectively leverages semantic and geometry information to guide exploration based on the environment's semantic distribution.

C. Target Identification for ZSON

The success rate of ZSON is directly influenced by accurate target identification, typically using pre-trained detection and
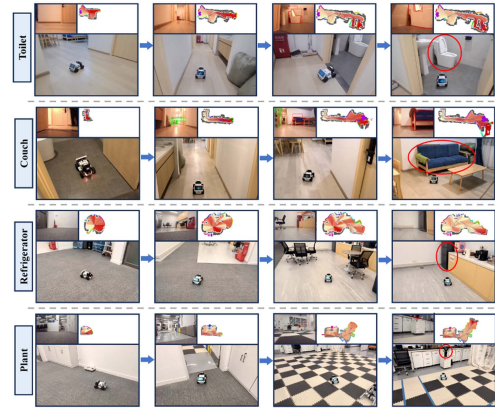

Fig. 1. Real-world Demonstration of ApexNav. We test ApexNav on various object goals in different environments. The figure shows the navigation process from start to finish, all successfully locating the target objects.

segmentation models [13], [21], [29] on RGB inputs. As noted in [3], false positives from these detectors remain a major limitation. Some methods [7], [27] rely solely on single-frame detector outputs, making them vulnerable to failure from mispredictions. Several works [9], [28] construct semantic maps via max-confidence fusion, which is prone to high-confidence misdetections that are difficult to correct later. To mitigate this, TriHelper [30] and SG-Nav [26] introduce validation modules: the former leverages VLMs for refinement, while the latter combines multi-frame scores with 3D scene graphs. Although these methods improve reliability, they lack a robust fusion mechanism to effectively manage detector errors in complex environments. We propose a target-centric semantic fusion method that aggregates multi-frame observations using context-aware confidence updates, effectively filtering out occasional errors and improving robustness.

III. PROBLEM FORMULATION

We tackle the ObjectNav task [2], where an agent must navigate to a specified object (e.g., a "chair") in an unknown environment. The agent uses an egocentric RGB-D camera and an odometry sensor that provides its displacement and orientation relative to the start. The agent is required to efficiently navigate to the target object, minimizing path length and reaching it within a defined success distance.

IV. METHODOLOGY

The overview of ApexNav is shown in Fig. 2. The agent constructs a frontier map and a semantic score map (Section IV-A). Combining the object goal with the similar objects inferred by the LLM, a target-centric semantic fusion method maintains semantic memory and determine the presence of a reliable target (Section IV-C). If a reliable target is found, an object waypoint is generated as the goal. Otherwise, the agent executes an adaptive exploration strategy to select the most suitable frontier waypoint as the goal (Section IV-B). The selected waypoint is then passed to the navigation module, which uses action evaluation to guide movement (Section IV-D)

A. Environmental Mapping

1) *Frontier Map:* We construct the frontier map on a 2D probabilistic grid, where each cell is labeled as free, occupied, or unknown via raycasting. Depth images are converted to point
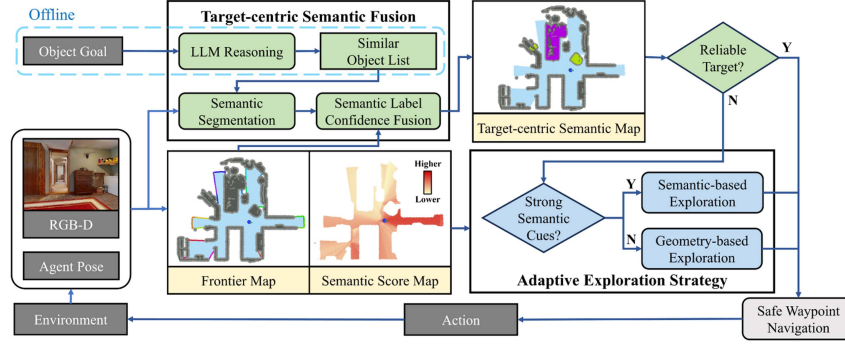
Fig. 2. System Architecture of ApexNav. Before the episode, an LLM offline generates a similar object list. The agent builds a frontier map, semantic score map, and a semantic map using target-centric fusion to identify reliable targets. If a reliable target is found, the safe waypoint navigation module directs the agent to it; otherwise, an adaptive exploration strategy selects a frontier waypoint for further exploration.

clouds and denoised using radius outlier removal. A frontier is defined as a free cell next to at least one unknown cell. Following [31], we incrementally update frontiers and cluster them using Principal Component Analysis, approximating each cluster by its center to simplify computation.

2) Semantic Score Map: Following [27], we build a 2D semantic score map to show the relevance between the environment and the target object, where higher scores mean stronger correlation. Each RGB image and a text prompt are fed into the pre-trained VLM BLIP-2 [10], which outputs a cosine similarity score using image-text matching.

We use objects typically associated with specific rooms (e.g., beds in bedrooms) to augment text prompts, which enhances the BLIP-2's prediction performance. Prompts like '*Seems there is a <target_object> or a <target_room> ahead*' enable BLIP-2 make earlier predictions at longer ranges. Room names are inferred using LLMs. For objects not tied to specific rooms (e.g., plants), prompts only mention the object. Cosine scores are projected onto observed free grids, weighted by confidence based on viewing angle. Grids near the optical axis get higher confidence, decreasing with angular offset as $\cos^2(\frac{\theta}{\theta_{\text{FoV}}/2} \cdot \frac{\pi}{2})$, where θ is the angular offset and θ_{FoV} is the FoV angle. To integrate semantic information across frames, the updated semantic score $s_{i,j}^{\text{new}}$ for grid (i,j) is computed as a confidence-weighted average: $s_{i,j}^{\text{new}} = \frac{c_{i,j}^{\text{cur}} s_{i,j}^{\text{cur}} + c_{i,j}^{\text{pre}} s_{i,j}^{\text{pre}}}{c_{i,j}^{\text{cur}} + c_{i,j}^{\text{pre}}}$, where $c_{i,j}^{\text{cur}}$ and $c_{i,j}^{\text{pre}}$ are the confidence scores from the current and previous frames, respectively. The updated confidence score is computed as $c_{i,j}^{\text{new}} = \frac{(c_{i,j}^{\text{cur}})^2 + (c_{i,j}^{\text{pre}})^2}{c_{i,j}^{\text{cur}} + c_{i,j}^{\text{pre}}}$, which emphasizes more confident predictions via squared weighting.

B. Adaptive Exploration Strategy

We propose an adaptive exploration strategy that adjusts its behavior based on the availability of semantic cues, ensuring more efficient use of environmental information by leveraging semantic reasoning when cues are strong and relying on geometric exploration when they are weak.

1) Mode-switching Criteria: To decide the exploration mode, we analyze the semantic score distribution across frontier clusters. Specifically, each frontier is assigned a score based on the semantic score map (Section IV-A2). We then compute the max-to-mean ratio $r = s_{\max}/\bar{s}$, where $s_{\max}$ and $\bar{s}$ are the maximum and mean scores, respectively. We also calculate the standard deviation $\sigma = \sqrt{\frac{1}{N_f} \sum_{i=1}^{N_f} (s_i - \bar{s})^2}$, where N_f is the number of frontier clusters. A high r suggests that some frontiers

are much more semantically relevant, pointing to promising areas. A high σ also indicates uneven semantic relevance and helps reduce outlier influence, providing a more reliable view of the distribution.

When both r and σ exceed their respective thresholds (r_t, σ_t), the agent switches to semantic-based exploration, prioritizing high-scoring frontiers to locate the target efficiently. Otherwise, it defaults to geometry-based exploration to expand coverage and gather more environmental information. This adaptive mode-switching mechanism balances rapid geometric exploration with semantic exploration, enhancing the efficiency of the object navigation process.

2) Semantic-based Exploration: When high-score frontiers are present, the system switches to semantic-based exploration. However, always choosing the highest-score frontier can lead to oscillations and inefficient backtracking, especially when scores are similar. To address this, we select a subset of high-score frontiers and solve a Traveling Salesperson Problem (TSP) to plan an optimal visiting sequence.

We model this problem as an Asymmetric TSP, which can be efficiently solved using the Lin-Kernighan-Helsgaun solver (LKH-Solver) [8], by appropriately designing the cost matrix M_{tsp}. Let N_h represent the number of high-score frontiers, with their center positions denoted by $\mathbf{p}_i$ $(i = 1, 2, \ldots, N_h)$, and the agent's position as p_0. The cost matrix M_{tsp} is a $N_h + 1$ dimensional square matrix, where the primary $(N_h + 1) \times N_h$ block contains the navigation distances $D(i,j)$ between each frontier and the agent, defined as $D(i,j) = \mathbf{Len}(\text{Path}(\mathbf{p}_i, \mathbf{p}_j))$. Here, $\text{Path}(\mathbf{p}_i, \mathbf{p}_j)$ represents the navigation path between $\mathbf{p}_i$ and $\mathbf{p}_j$, computed using the A* algorithm, and $\mathbf{Len}(\text{Path}(\cdot))$ denotes the total length of this path. The return cost to the agent is set to zero. The cost matrix $\mathbf{M_{tsp}}(i,j)$ is defined as:

$$\mathbf{M_{tsp}}(i,j) = \begin{cases} 0 & \text{if } j = 0, \\ D(i,j) & \text{if } i \in [0, N_h], j \in (0, N_h]. \end{cases} \quad (1)$$

Finally, The LKH-Solver computes the optimal path visiting all high-score frontiers, as shown in Fig. 3. This approach balances semantic relevance with global path efficiency, reducing redundancy and enhancing exploration consistency.

3) Geometry-based Exploration: When frontier semantic scores are highly uniform, the system switches to geometry-based exploration to quickly acquire new information and expand coverage. As shown in Section V-D, the nearest-frontier strategy often outperformed TSP-based planning. Although the result was initially unexpected because TSP is typically more efficient in full-coverage tasks [31], further analysis revealed that
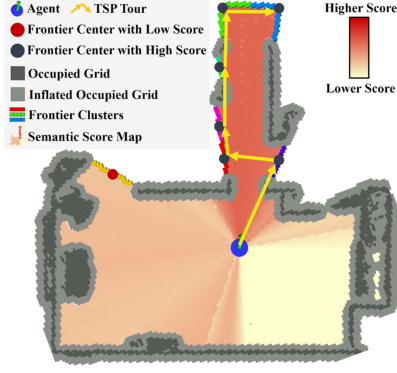
Fig. 3. Illustration of Semantic-based Exploration. Low-score frontier clusters (e.g., the one on the left in the figure) are excluded. A TSP tour is computed over the remaining high-score frontiers, and the agent chooses the first one on the tour as the next navigation goal.

ObjectNav focuses on locating the target rather than performing exhaustive exploration. In this setting, the greedy strategy better aligns with the task objective, leading us to adopt it for geometry-based exploration. Fig. 4 illustrates how our method adaptively switches modes to efficiently locate the target while avoiding irrelevant regions.

C. Target-centric Semantic Fusion

We introduce an object-level semantic fusion method that focuses on the target object and similar ones, improving the reliability of target recognition, as illustrated in Fig. 5.

1) Object Reasoning with LLMs: We leverage the reasoning ability of LLMs with Chain-of-Thought prompting [23] to identify objects that are easily misclassified as the target. Considering that detector performance can be affected by object size, as small objects often yield low confidence, the LLM is also prompted to generate an adaptive confidence threshold c_{th} for the target. An object is treated as a reliable target only if its fused confidence exceeds this threshold. The agent then navigates to the object. This process yields a comprehensive object list L_{obj} with M_{obj} entries, including the target and similar objects, guiding subsequent decisions.

2) Object Detection and Segmentation: The RGB image and the text descriptions of all objects to be detected L_{obj} are processed by the detector. We use YOLOv7 [21] for objects within the COCO dataset due to its strong performance, and Grounding-DINO [13] for the rest. Detected object instances are segmented by Mobile-SAM [29] using their bounding boxes, and their corresponding point clouds are extracted from the depth map. DBSCAN is applied to remove noise. From a single frame, we obtain N_{det} detected objects, represented as $I_{det} = \{(pt^i_{det}, c^i_{det}, l^i_{det}) \mid i = 1, 2, \ldots, N_{det}\}$, where pt^i_{det}, c^i_{det}, and l^i_{det} denote the point cloud, confidence, and label of the i-th object, respectively.

3) Semantic Label Confidence Fusion: Our fusion method not only incorporates the detector's current detection results but also considers objects that were previously detected but are not identified within the agent's current FoV. To reduce computational overhead and align with our object-level objective, we use objects as the fundamental units for fusion.

Structure of Object Clusters: Each cluster consists of associated 2D grids and maintains object-level information $I_o = \{(pt^l_o, c^l_o, n^l_o) \mid l = 1, 2, \ldots, M_{obj}\}$, where M_{obj} is the number of object categories (see Sec. IV-C1), and l is the label index.

For each label, the cluster stores the 3D point cloud pt^l_o, confidence c^l_o, and accumulated detection volume n^l_o. This structure allows each cluster to retain multi-label information, offering a comprehensive representation of observed objects.

Object Clustering and Fusion: We fuse single-frame detection results I_{det} and update object clusters incrementally. Given an arbitrary detection $I^i_{det} = (pt^i_{det}, c^i_{det}, l^i_{det})$ from I_{det}, the clustering and fusion process proceeds as follows. The point cloud pt^i_{det} is projected onto the 2D grid map, and points landing in unoccupied grid cells are filtered out. If no existing object cluster intersects with the projected points, a new object cluster is created with M_{obj} empty entries, each corresponding to a different label. The entry for label l^i_{det} is directly filled with the information from I^i_{det}. Otherwise, I^i_{det} is fused into the intersecting cluster O_{last}, where the pre-fusion information for label l^i_{det} is $(pt_{last}, c_{last}, n_{last})$, and the post-fusion result becomes $(pt_{new}, c_{new}, n_{new})$. Our fusion process weights observations based on detection volume, approximated by the number of points in the down-sampled point cloud, as a larger detected area indicates a more reliable result. Specifically, pt^i_{det} is down-sampled to a consistent resolution r_{down}, and the resulting number of points in the point cloud, v^i_{det}, represents the detection volume for a single detection result. During fusion, pt^i_{det} is merged with $pt^{l^i_{det}}_{last}$, and the merged point cloud is down-sampled to generate $pt^{l^i_{det}}_{new}$. The values c^i_{det} and v^i_{det} are integrated into $c^{l^i_{det}}_{last}$ and $n^{l^i_{det}}_{last}$, updating the accumulated observed points $n^{l^i_{det}}_{new}$ and the confidence score $c^{l^i_{det}}_{new}$:

$$n^{l^i_{det}}_{new} = n^{l^i_{det}}_{last} + v^i_{det},$$

$$c^{l^i_{det}}_{new} = \frac{n^{l^i_{det}}_{last}}{n^{l^i_{det}}_{new}} \cdot c^{l^i_{det}}_{last} + \frac{v_{det}}{n^{l^i_{det}}_{new}} \cdot c^i_{det}. \qquad (2)$$

We also consider object clusters in the current FoV that are missing from the current detection. Such absence may suggest incorrect associations, leading to a confidence penalty. For each label in such clusters, the detection confidence c'_{det} is set to 0. The fusion weight of this detection is also influenced by the detection volume v'_{det}, determined by the number of points in the point cloud of each label within the object cluster that have a nearest neighbor within the threshold r_{down} in the point cloud derived from the current observation's depth image. c'_{det} and v'_{det} are fused using (2) for each label within the object cluster, effectively lowering the confidence. This approach effectively mitigates abnormal confidence scores caused by occasional false positives, as illustrated in Fig. 6(b).

Determine the Best-matching Label: After fusion, each object cluster contains information for multiple labels. We select the label with the highest $c_o \cdot n_o$ as the best match, where c_o is the confidence score and n_o is the detection volume, balancing reliability and observation count.

4) Selection of Reliable Targets: Only object clusters whose best-matching label matches the target are considered. Those with confidence above c_{th} are treated as *reliable targets*, and the agent navigates to them while continuing fusion. Clusters below the threshold are marked as *suspected targets* and are only considered after no frontiers to explore, with the most confident one selected as the goal.

D. Safe Waypoint Navigation

After initialization, the agent selects either a frontier or an object waypoint as the navigation goal, depending on whether a reliable target is present. Under the Habitat setting, the agent
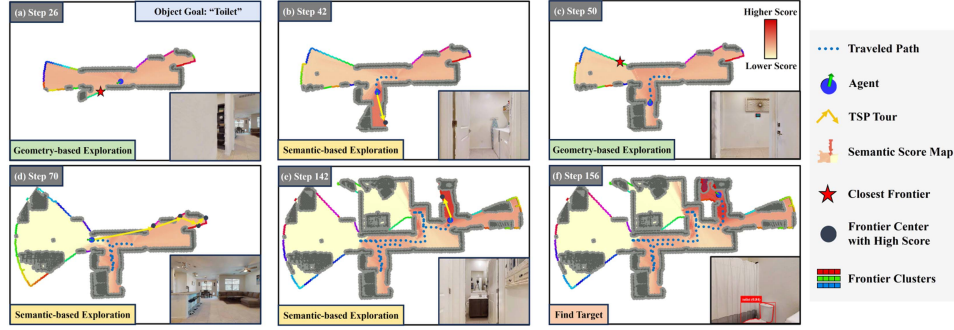
Fig. 4. An Illustrative Example of Adaptive Exploration. The agent starts by rotating to initialize the environment. At step 26, it starts searching for the target "toilet." Similar semantic scores on both sides prompt a switch to geometry mode, selecting the nearest frontier. At step 42, it detects washing machines and infers a likely nearby toilet, switching to semantic mode to explore further. When no toilet is found, it returns to geometry mode at step 50 as semantic scores even out. At step 70, it identifies a living room ahead, unlikely to contain a toilet, and switches to semantic mode to explore the high-scoring region on the right. At step 142, it detects a likely bathroom, focuses on the area, and successfully finds the toilet at step 156.
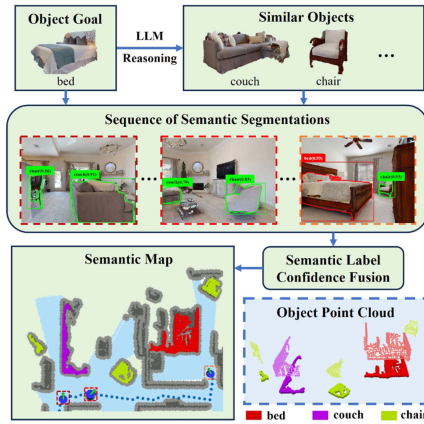

Fig. 5. Pipeline of Target-centric Semantic Fusion. The agents, enclosed by different-colored dashed boxes, represent semantic segmentations at different time steps. The blue dashed box in the bottom right shows the 3D point cloud information of different objects in the 2D semantic map.

(a) Case 1: The target is a bed. At step 3, the detector misclassifies a couch as the bed. However, as the object cluster was already fused twice in steps 1 and 2 with higher-confidence couch data, the misclassification has no effect.

(b) Case 2: The target is a bed. The agent misclassifies a table as a bed in step 1. With its confidence score exceeding the reliable threshold, the agent navigates to the target. However, the object is not detected in steps 2 and 3, the confidence drops below the threshold, so the agent abandons the target.
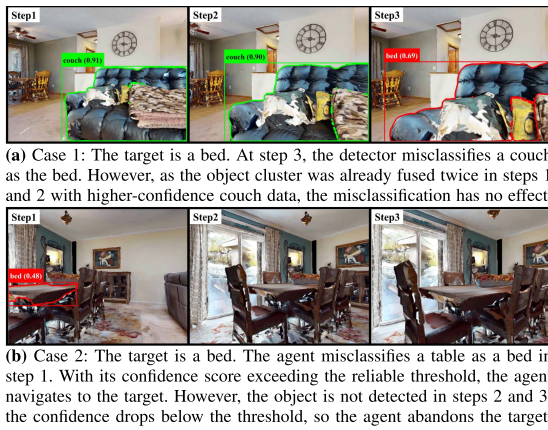
Fig. 6. Two Cases Solvable by Target-centric Semantic Fusion.

must choose from a set of discrete actions. Many previous methods [5], [9], [26] compute the shortest path and follow it directly; however, these paths often run too close to obstacles, increasing the risk of collisions. To address this, we evaluate actions by jointly considering both efficiency and safety.

1) Efficiency Cost: We use A* to find the shortest path P and select the first point beyond d_{local} as the local target p_{local}. Two cost terms guide action selection: the distance cost $cost_{target}$,

which encourages moving toward p_{local}:

$$cost_{target} = \omega_t \cdot \|p_{new} - p_{local}\|. \quad (3)$$

The proximity change cost $cost_{prox}$ penalizes actions that increase the distance to p_{local}:

$$cost_{prox} = \omega_p \cdot (\|p_{new} - p_{local}\| - \|p_{cur} - p_{local}\|) \cdot$$
$$\cdot \mathbb{1}_{\|p_{new} - p_{local}\| > \|p_{cur} - p_{local}\|} \quad (4)$$

Here, $\mathbb{1}_{\{\cdot\}}$ denotes the indicator function, which returns 1 if the condition in the subscript is met and 0 otherwise.

2) Safety Cost: To evaluate action safety, we build a local ESDF map and compute a safety cost $cost_{safe}$ by sampling K points along the action path:

$$cost_{safe} = \sum_{i=1}^{K} \frac{\omega_s}{d_{obs}(p_{sa_i}) + \epsilon} \cdot \mathbb{1}_{d_{obs}(p_{sa_i}) < d_{th}}. \quad (5)$$

Here, $d_{obs}(p_{sa_i})$ is the distance to the nearest obstacle. Only points within a threshold d_{th} add to the cost, penalizing unsafe actions. ϵ avoids division by zero, and ω_s scales the cost.

Finally, the action with the lowest total cost is selected as the agent's next move, ensuring efficient and safe navigation. Notably, in real-world settings where discrete actions are no longer applicable, we employ MINCO [22] to generate a safe and continuous spatiotemporal trajectory.

V. EXPERIMENTS

A. Simulation Experimental Setup

Datasets: We evaluate methods in Habitat [16] simulator on three large-scale datasets: *HM3Dv1* (HM3D-Semantics-v0.1 [18] from 2022 Habitat Challenge, 2000 episodes, 20 scenes, 6 goal categories), *HM3Dv2* (HM3D-Semantics-v0.2 [24] from 2023 Habitat Challenge, 1000 episodes, 36 scenes, 6 goal categories), and *MP3D* (Matterport3D [4] from 2021 Habitat Challenge, 2195 episodes, 11 scenes, 21 goal categories).

Evaluation Metrics: We report Success Rate (SR) and Success weighted by inverse Path Length (SPL) for all methods, where SR measures episode success and SPL assesses efficiency relative to the optimal path (SPL is 0 for failures). For ablation studies, we also report SoftSPL, which accounts for goal-directed progress in failed episodes. Higher SR, SPL, and SoftSPL indicate superior performance.

Implementation Details: In the Habitat setting, the agent selects actions from a discrete set $\mathcal{A}$: MOVE_FORWARD (0.25 m), TURN_LEFT (30°), TURN_RIGHT (30°), LOOK_UP (30°), LOOK_DOWN (30°), and STOP. Each episode is limited to

TABLE I
BENCHMARK COMPARISONS. DATA WITH A STATEBLUE BACKGROUND SHOWS THE BEST RESULT, WHILE LIGHTBLUE BACKGROUND INDICATES THE SECOND-BEST.

Method	Zero-shot	HM3Dv1		HM3Dv2		MP3D	
		SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
SemExp[5]	×	–	–	–	–	36.0	14.4
PONI[19]	×	–	–	–	–	31.8	12.1
ZSON[15]	×	25.5	12.6	–	–	15.3	4.8
CoW[7]	√	–	–	–	–	7.4	3.7
ESC[32]	√	39.2	22.3	–	–	28.7	14.2
L3MVN[28]	√	50.4	23.1	36.3	15.7	34.9	14.5
OpenFMNav[9]	√	54.9	24.4	–	–	37.2	15.7
InstructNav[14]	√	–	–	58.0	20.9	–	–
VLFM[27]	√×	52.5	30.4	63.6	32.5	36.4	17.5
VLFM*[27]	√	50.9	23.6	56.9	27.5	32.5	15.9
TriHelper[30]	√	56.5	25.3	–	–	–	–
SG-Nav[26]	√	54.0	24.9	49.6	25.5	40.2	16.0
ApexNav(Ours)	√	**59.6**	**33.0**	**76.2**	**38.0**	39.2	**17.8**

500 steps. The agent operates within a perception range of $[0.0\,\text{m}, 5.0\,\text{m}]$, with a success distance of $0.2\,\text{m}$. The onboard camera is mounted at a height of $0.88\,\text{m}$. For semantic segmentation, we use YOLOv7 [21], GroundingDINO [13], and Mobile-SAM [29]. The semantic score map is generated by BLIP-2 [10]. DeepSeek-V3 [11] is used to generate the similar object list, rooms where objects are likely to appear, and detection confidence thresholds. The parameters are set as follows: $r_t = 1.10$, $\sigma_t = 0.015$ (Section IV-B1), and $d_{\text{local}} = 0.75$, $\omega_t = 200, \omega_p = 2000, \omega_s = 1.0, \epsilon = 10^{-2}, K = 10, d_{\text{th}} = 0.15$ (Section IV-D). All simulation experiments are run on a system with an NVIDIA GeForce RTX 4090 GPU and an Intel i9-13900K ×32 CPU.

Baselines: SemExp [5], PONI [19], and ZSON [15] rely on task-specific training, limiting their generalization in zero-shot settings. CoW [7] uses a nearest-frontier exploration strategy without semantic reasoning. ESC [32], L3MVN [28], TriHelper [30], and OpenFMNav [9] improve exploration by building semantic maps and using LLMs to select frontiers. VLFM [27] and InstructNav [14] use LLMs or VLMs to create value maps that guide agents. VLFM* replaces the pre-trained PointNav module with a shortest-path planner for fair comparison. SG-Nav [26] further enhances semantic reasoning by prompting LLMs with 3D scene graphs to select frontiers.

B. Comparison with State-of-the-art

Table I compares ApexNav with SOTA methods on HM3Dv1, HM3Dv2, and MP3D datasets. Blank cells indicate that the method was not evaluated on that dataset. As only InstructNav reported HM3Dv2 results, we re-evaluated several open-source methods under our settings for fairness. ApexNav demonstrates superior performance over all the SOTA methods. On HM3Dv1, ApexNav achieves a 5.5% improvement in SR compared to the suboptimal TriHelper and an 8.6% enhancement in SPL relative to VLFM. On HM3Dv2, ApexNav exhibits a 19.8% increase in SR and a 16.9% improvement in SPL compared to the previously best-performing method, VLFM. On MP3D, while the SR of ApexNav is marginally lower than that of SG-Nav, it achieves an 11.3% higher SPL compared to SG-Nav and a 1.7% improvement compared to suboptimal VLFM. The great enhancement in SR validates the efficacy of the target-centric semantic fusion mechanism introduced by ApexNav in rectifying error

detection, while the notable improvement in SPL is predominantly attributable to the adaptive exploration strategy's superior efficiency and robustness in exploring different environments.

C. Failure Cause Analysis

Task outcomes are categorized as follows:
- *Success (A):* The agent stops within $0.2\,\text{m}$ of the goal and the simulator confirms success.
- *Different Floor (B):* All target objects are on a different floor from the agent's start position.
- *False Positive (C):* The agent stops near a detected object, but the simulator returns failure, meaning the wrong object was found.
- *No Frontier (D):* The agent runs out of frontiers to explore without passing the target.
- *Stepout (E):* The agent exceeds the 500-step limit without passing the target.
- *Missing Target (F):* Passes the target but fails to recognize it, either due to no frontiers left or step limit reached.

As shown in Fig. 7, "False Positive" cases account for over 11% across all datasets, mainly due to persistent detection errors and incorrect fusion. Another major cause is annotation issues, where the agent finds the target but the task still fails because the object is not labeled.

Except for the HM3Dv2 dataset, which mainly consists of single-floor tasks, the others show many "Different Floor" failures (> 13%). This is because ApexNav builds only a single-layer 2D map, so such cross-floor tasks are treated as failures by default. This limitation also leads to many "No Frontier" failures. When the agent starts at a staircase between two floors, the vertical distance to the target is not large enough to be recognized as on a different floor. However, since the agent builds a 2D map centered at its starting position, it may find no frontiers to explore after initialization, resulting in task failure. This limitation is mainly an engineering constraint, and could be addressed by incorporating 3D maps in future work.

"Stepout" failures are more common in HM3Dv2 and MP3D due to their larger scene scale, which make it hard for the agent to find the target within 500 steps. Raising the step limit could help. Additionally, some scenes in MP3D have black holes that cause missing depth data, leading to map mismatches and the agent getting stuck. We recommend focusing on datasets with more complete scene structures, like HM3Dv2, for future research. "Missing Target" cases are rare and mostly involve small or ambiguous objects like plants, clothes, or pictures, which are hard to detect. More robust detection and segmentation models are needed to address this.

D. Ablation Study on Adaptive Exploration Strategy

1) Ablation Study on Exploration Strategy: We conduct an ablation study on HM3Dv2 to evaluate how different exploration strategies affect performance. Four strategies were tested: **(a)***TSP-Semantic*: selects high-scoring frontiers and solves TSP to plan the order; **(b)***Greedy-Semantic*: chooses the frontier with the highest semantic score; **(c)***TSP-Geometry*: solves TSP over all frontiers; **(d)***Greedy-Geometry*: selects the nearest frontier. **(e)***Utility-based*: selects the highest utility score (semantic score divided by path length [6]). We evaluate single-stage strategies (using one method throughout), two-stage strategies (switching between semantic and geometric modes based on thresholds $\sigma_t = 0.015$ and $r_t = 1.10$) and utility-based strategies. Results are shown in Table II.
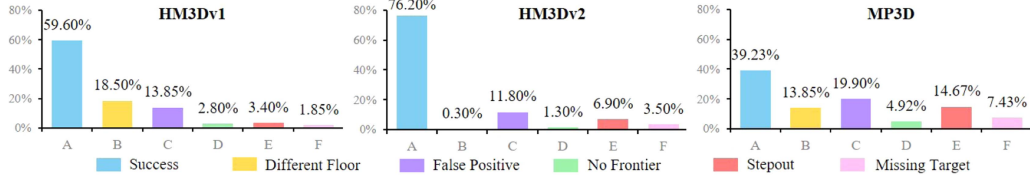
Fig. 7.　Failure Cause Statistics of ApexNav on the Three Datasets.

TABLE II
ABLATION STUDY ON EXPLORATION STRATEGY.

Exploration Strategy		SR↑	SPL↑	SoftSPL↑
Semantic	Geometry			
Greedy	–	72.90	36.07	37.85
TSP	–	74.50	37.03	39.02
–	Greedy	73.80	36.42	38.40
–	TSP	72.90	34.25	36.87
TSP	TSP	74.20	36.79	38.88
Greedy	Greedy	74.80	37.67	39.65
Greedy	TSP	75.00	36.61	38.61
TSP	Greedy	**76.20**	**38.03**	**40.44**
Utility-based		74.40	36.47	38.94

Fig. 8.　Ablation Study on Exploration Metrics.

TABLE III
ABLATION STUDY ON EXPLORATION METRICS.

Serial	Exploration Metrics	SR↑	SPL↑	SoftSPL↑
i	$\sigma_t = 0.000, r_t = 1.00$	74.50	37.03	39.02
ii	$\sigma_t = 0.010, r_t = 1.05$	73.70	37.80	39.97
iii	$\sigma_t = 0.015, r_t = 1.10$	**76.20**	**38.03**	**40.44**
iv	$\sigma_t = 0.020, r_t = 1.15$	75.40	37.76	40.17
v	$\sigma_t = 0.025, r_t = 1.20$	74.90	37.42	40.05
vi	$\sigma_t = \infty, r_t = \infty$	73.80	36.42	38.40

TABLE IV
ABLATION STUDY ON SEMANTIC FUSION.

Method	HM3Dv1		HM3Dv2		MP3D	
	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
(a) w/o Similar Objects	56.95	30.40	75.00	37.63	28.70	15.43
(b) w/o Fusion	44.50	25.48	55.00	29.72	27.84	14.03
(c) w/o Fusion (High DCT)	55.20	31.24	68.20	34.99	37.81	17.16
(d) w/o Observation	50.48	29.17	70.20	37.44	35.58	16.09
(e) w Max-fusion	51.43	28.97	67.90	37.04	32.11	15.32
Ours	**59.60**	**32.95**	**76.20**	**38.03**	**39.23**	**17.81**

The results show that two-stage strategies outperform single-stage ones, highlighting the advantage of adaptively switching between semantic and geometric exploration. Among the single-stage strategies, TSP-Semantic and Greedy-Geometry performed best, and their combination achieves the highest overall score: Greedy-Geometry is efficient when semantic cues are weak, while TSP-Semantic guides the agent effectively once strong semantic cues emerge. Meanwhile, our strategy outperforms utility-based methods by better prioritizing semantically informative regions, even when they are far away, unlike utility-based methods that often undervalue such areas due to longer path costs.

2) Ablation Study on Exploration Metrics: To explore the balance between semantic-based and geometry-based exploration, we test various combinations of σ_t and r_t, using SR, SPL, and SoftSPL as evaluation metrics. The results show a hump-shaped curve (see Table III and Fig. 8). The best performance is achieved with $\sigma_t = 0.015$ and $r_t = 1.10$, supporting the idea that relying too much on either semantic or geometric information reduces efficiency. Instead, an adaptive balance between the two is key to effective exploration.

E. Ablation Study on Target-centric Semantic Fusion

To assess the impact of the target-centric semantic fusion method, we conduct many ablation experiments. The following control groups were designed: (**a**)*w/o Similar Objects*: Removes the auxiliary detection module for similar objects; (**b, c**)*w/o Fusion*: Removes the semantic fusion module and directly overwrites cluster confidence using detection results; (**b**) uses
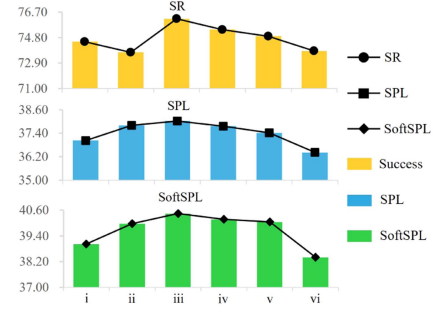
a detector confidence threshold (DCT) as in ApexNav, while (**c**) uses a higher DCT as in VLFM. A lower DCT makes the detector more prone to false positives. (**d**)*w/o Observation*: The confidence of the object cluster is not reduced by fusing low-confidence detection results when no objects are detected; (**e**)*w Max-fusion*: Replaces ApexNav's fusion method with max-confidence fusion. The results are presented in Table IV.

Both the (**b**) and (**c**) groups underperform compared to Ours, emphasizing the importance of fusion. Although (**c**) reduces false detections with a higher DCT, it also misses some correct ones. ApexNav uses a lower DCT and refines more candidates through fusion, leading to better results and showing stronger robustness to false detections. Additionally, ApexNav outperforms the max-fusion approach (**e**), proving its effectiveness in preventing task failures caused by high-confidence mislabeling. The other two experiments emphasize the significance of fusing similar objects and reducing confidence for objects that were previously detected but are currently undetected. Overall, the target-centric semantic fusion method delivers the best performance, demonstrating its effectiveness.

F. Real-world Deployment

We conduct real-world experiments using the AgileX LIMO robot. A laptop equipped with an RTX 4060 Laptop GPU and an AMD R9 7945HX CPU served as the ROS master and communicated with the robot via a ROS-based master-slave architecture. We use MINCO [22] to generate collision-free spatiotemporal trajectories for waypoint navigation. ApexNav is evaluated in dormitory and laboratory environments across

TABLE V
TIME CONSUMPTION OF MODULES ON DIFFERENT DEVICES. THE *DESIRED INTERVAL* INDICATES THE RUNTIME RANGE REQUIRED FOR STABLE REAL-TIME PERFORMANCE. (A) RUNTIME OF MODULES ON DIFFERENT GPU DEVICES. (B) RUNTIME OF MODULES ON DIFFERENT CPU DEVICES.

Modules	GPU Runtime (ms)		Desired Interval (ms)
	RTX 4060 Laptop	RTX 4090	
Detection+Segmentation	[190, 320]	[65, 125]	250
BLIP2	[160, 190]	[50, 55]	200

(a) Runtime of modules on different GPU devices.

Modules	CPU Runtime (ms)		Desired Interval (ms)
	AMD Ryzen9 7945hx	Intel i9-13900K	
Semantic Object Map	[15, 20]	[5, 15]	250
Semantic Score Map	[1, 2]	[0.5, 1]	200
Grid Map	[30, 50]	[20, 35]	100
ESDF Map	[1, 3]	[1, 2]	100
Exploration Planning	[50, 200]	[40, 150]	500

(b) Runtime of modules on different CPU devices.

TABLE VI
QUANTITATIVE STATISTICS IN REAL-WORLD EXPERIMENTS RESULTS ARE AVERAGED OVER ALL TRIALS. TD: TRAVELED DISTANCE; TT: TRAVELED TIME.

Trials	SR (%)	SPL	TD (m)	TT (s)
6	100	0.723	15.85	56

six trials, targeting toilet, couch, and chair in the former, and refrigerator, plant, and TV in the latter. Representative results and quantitative statistics are shown in Fig. 1 and Table VI. Module runtimes are summarized in Table V. Most modules run within the desired interval, even on a consumer-grade laptop. Although detection slightly exceed this range, it has little effect on overall task performance. This further highlights the efficiency and deployability in practical scenarios.

VI. CONCLUSION

In this letter, we propose ApexNav, an efficient and reliable ZSON framework. ApexNav features an adaptive exploration strategy that dynamically combines semantic- and geometry-based approaches, enabling the agent to effectively balance between them based on the reliability of semantic cues. In addition, we propose a target-centric semantic fusion method that preserves long-term memory of the target object and its semantically similar counterparts, effectively mitigating task failures caused by misidentification and significantly improving success rates. ApexNav achieves SOTA performance on three benchmark datasets and is further validated through extensive real-world experiments.

While ApexNav performs well, it has several limitations worth exploring. First, it assumes the target is visible, but objects may be hidden (e.g., in drawers), requiring interaction or manipulation. In real-world tests, we also observed two issues. One is that the system assumes all areas in the sensor's view are fully explored, which may not hold for small objects that require finer-grained exploration. Another is that when the target has weak semantic relevance to its surroundings, semantic guidance becomes ineffective, leading to inefficient geometric exploration. Using richer language priors, as in Vision-Language Navigation, may help overcome this.

REFERENCES

[1] J. Achiam et al., "GPT-4 technical report," 2023, *arXiv:2303.08774*.
[2] D. Batra et al., "ObjectNAV revisited: On evaluation of embodied agents navigating to objects," 2020, *arXiv:2006.13171*.
[3] F. L. Busch et al., "One map to find them all: Real-time open-vocabulary mapping for zero-shot multi-object navigation," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2025, pp. 14835–14842.
[4] A. Chang et al., "MatterPort3D: Learning from RGB-D data in indoor environments," in *Proc. Int. Conf. 3D Vis. (3DV)*, 2017, pp. 667–676.
[5] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 4247–4258.
[6] J. Chen, G. Li, K. Suryansh, G. Bernard, and Y. Fisher, "How to not train your dragon: Training-free embodied object goal navigation with semantic frontiers," in *Proc. Robot.: Sci. Syst.*, 2023, p. 75.
[7] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 23171–23181.
[8] K. Helsgaun, "An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems," *Roskilde: Roskilde Univ.*, vol. 12, pp. 966–980, 2017.
[9] Y. Kuang, H. Lin, and M. Jiang, "OpenFMNAV: Towards open-set zero-shot object navigation via vision-language foundation models," 2024, *arXiv:2402.10670*.
[10] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 19730–19742.
[11] A. Liu et al., "DeepSeek-V3 technical report," 2024, *arXiv:2412.19437*.
[12] R. Liu et al., "Handle object navigation as weighted traveling repairman problem," 2025, *arXiv:2503.06937*.
[13] S. Liu et al., "Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection," in *Proc. Eur. Conf. Comput. Vis.*, Berlin, Germany, 2024, pp. 38–55.
[14] Y. Long, W. Cai, H. Wang, G. Zhan, and H. Dong, "InstructNAV: Zero-shot system for generic instruction navigation in unexplored environment," in *Proc. Conf. Robot Learn.*, 2025, pp. 2049–2060.
[15] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra, "ZSON: Zero-shot object-goal navigation using multimodal goal embeddings," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 32340–32352.
[16] X. Puig et al., "Habitat 3.0: A co-habitat for humans, avatars and robots," 2023, *arXiv:2310.13724*.
[17] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.
[18] S. K. Ramakrishnan et al., "Habitat-MatterPort 3 D dataset (HM3D): 1000 large-scale 3 D environments for embodied AI," 2021, *arXiv:2109.08238*.
[19] S. K. Ramakrishnan et al., "Potential functions for objectgoal navigation with interaction-free learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18890–18900.
[20] H. Touvron et al., "Llama: Open and efficient foundation language models," 2023, *arXiv:2302.13971*.
[21] C.-Y. Wang, A. Bochkovskiy, and H.-Y. Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. pattern Recognit.*, 2023, pp. 7464–7475.
[22] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE Trans. Robot.*, vol. 38, no. 5, pp. 3259–3278, Oct. 2022.
[23] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," *Proc. Adv. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 24824–24837.
[24] K. Yadav et al., "Habitat-MatterPort D semantics dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 4927–4936.
[25] X. Ye and Y. Yang, "Efficient robotic object search via HIEM: Hierarchical policy learning with intrinsic-extrinsic modeling," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4425–4432, Jul. 2021.
[26] H. Yin, X. Xu, Z. Wu, J. Zhou, and J. Lu, "SG-NAV: Online 3 D scene graph prompting for LLM-based zero-shot object navigation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, vol. 37, pp. 5285–5307.
[27] 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. Automat.*, Yokohama, Japan, 2024, pp. 42–48.
[28] B. Yu, H. Kasaei, and M. Cao, "L3MVN: Leveraging large language models for visual target navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Detroit, MI, USA, 2023, pp. 3554–3560.
[29] C. Zhang et al., "Faster segment anything: Towards lightweight sam for mobile applications," 2023, *arXiv:2306.14289*.
[30] L. Zhang et al., "Zero-shot object navigation with dynamic assistance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Abu Dhabi, United Arab Emirates, 2024, pp. 10035–10042.
[31] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "FUEL: Fast UAV exploration using incremental frontier structure and hierarchical planning," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 779–786, Apr. 2021.
[32] K. Zhou et al., "Exploration with soft commonsense constraints for zero-shot object navigation," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 42829–42842.