Stairway to Success: An Online Floor-Aware Zero-Shot Object-Goal Navigation Framework via LLM-Driven Coarse-to-Fine Exploration

Zeying Gong, *Student Member, IEEE*, Rong Li, Tianshuai Hu, Ronghe Qiu, *Graduate Student Member, IEEE*, Lingdong Kong, Lingfeng Zhang, Guoyang Zhao, Yiyi Ding, and Junwei Liang, *Member, IEEE*

***Abstract*—Deployable service and delivery robots struggle to navigate multi-floor buildings to reach object goals, as existing systems fail due to single-floor assumptions and requirements for offline, globally consistent maps. Multi-floor environments pose unique challenges including cross-floor transitions and vertical spatial reasoning, especially navigating unknown buildings. Object-Goal Navigation benchmarks like HM3D and MP3D also capture this multi-floor reality, yet current methods lack support for online, floor-aware navigation. To bridge this gap, we propose *ASCENT*, an online framework for Zero-Shot Object-Goal Navigation that enables robots to operate without pre-built maps or retraining on new object categories. It introduces: 1) a Multi-Floor Abstraction module that dynamically constructs hierarchical representations with stair-aware obstacle mapping and cross-floor topology modeling, and 2) a Coarse-to-Fine Reasoning module that combines frontier ranking with LLM-driven contextual analysis for multi-floor navigation decisions. We evaluate on HM3D and MP3D benchmarks, outperforming state-of-the-art zero-shot approaches, and demonstrate real-world deployment on a quadruped robot.**

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

I. Introduction

MODERN robots are expected to operate in multi-floor buildings, from homes to offices. A simple command like *"Find the TV"* becomes challenging when the target is on another floor, requiring cross-floor navigation and vertical reasoning in unexplored buildings. This highlights a critical gap in robotics: **Object-Goal Navigation (OGN)** in multi-floor settings. However, existing online navigation systems often fail in these scenarios, limiting their real-world applications.

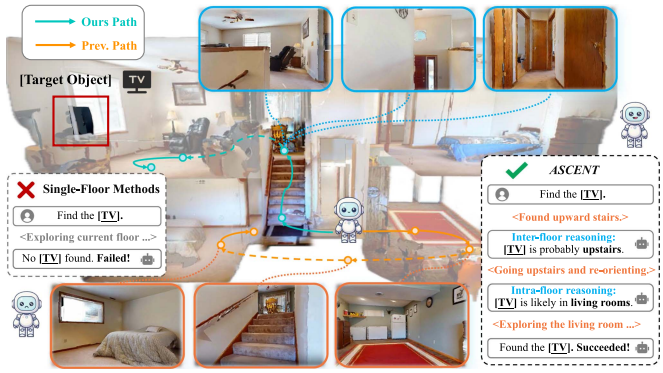


Fig. 1. **Motivation of *ASCENT*.** Unlike prior approaches that fail in multi-floor scenarios, our method enables online multi-floor navigation. By reasoning across floors, our policy succeeds in locating the goal and demonstrates a meaningful step forward in ZS-OGN.

To quantify multi-floor navigation's importance, we analyze two widely-used OGN benchmarks, HM3D [1] and MP3D [2]. Over half of validation episodes involve multi-floor buildings, with up to 28% requiring explicit floor transitions (see Section IV-A). However, most OGN methods assume single-floor use. While multi-floor SLAM systems [3] improve localization, they lack semantic reasoning for OGN in unseen scenes.

Moreover, recent OGN methods are either learning-based [4], [5], [6], [7], requiring retraining for unseen environments, or **Zero-Shot (ZS-OGN)** [8], [9], [10], [11], [12], [13], which generalize without retraining but neglect cross-floor planning. Even multi-floor methods like MFNP [14] face limitations in spatial overlap, one-way transitions and heuristic thresholds. This motivates our online multi-floor navigation method as shown in Fig. 1.

Beyond multi-floor navigation, efficient planning poses another challenge for ZS-OGN. Large Language Model (LLM) planners [10], [11], [12], [13] enable high-level reasoning but incur high costs and latency, while Vision-Language Model (VLM) methods like VLFM [9] are faster but lack global planning, causing oscillation and suboptimal paths.

Received 11 September 2025; accepted 28 December 2025. Date of publication 19 January 2026; date of current version 27 January 2026. This article was recommended for publication by Associate Editor Z. Wu and Editor P. Vasseur upon evaluation of the reviewers' comments. This work was supported by Guangzhou Municipal Science and Technology Project under Grant 2024A03J0619. *(Corresponding author: Junwei Liang.)*

Zeying Gong, Rong Li, Ronghe Qiu, Guoyang Zhao, and Yiyi Ding are with The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511453, China (e-mail: zgong313@connect.hkust-gz.edu.cn; rongli@hkust-gz.edu.cn; rqiu683@connect.hkust-gz.edu.cn; gzhao492@connect.hkust-gz.edu.cn; ydingaz@connect.hkust-gz.edu.cn).

Tianshuai Hu is with The Hong Kong University of Science and Technology, Hong Kong, SAR 999077, China (e-mail: thuaj@connect.ust.hk).

Lingdong Kong is with the National University of Singapore, Singapore 119077 (e-mail: lingdong@comp.nus.edu.sg).

Lingfeng Zhang is with Tsinghua University, Beijing 100084, China (e-mail: lfzhang715@gmail.com).

Junwei Liang is with the The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511453, China, and also with the The Hong Kong University of Science and Technology, Hong Kong, SAR 999077, China (e-mail: junweiliang@hkust-gz.edu.cn).

The source code is available at https://github.com/Zeying-Gong/ascent.

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2026.3655265, provided by the authors.

Digital Object Identifier 10.1109/LRA.2026.3655265

To address these, we propose *ASCENT*, an online framework unifying dynamic multi-floor mapping with LLM-driven spatial reasoning and dataset-derived statistical priors for ZS-OGN. Our Multi-Floor Abstraction module that captures floor connectivity and enables inter-floor reasoning, allowing exploration of unseen buildings with semantic consistency across floors. Our Coarse-to-Fine Reasoning module reduces the need for frequent LLM calls-by over 90% compared to prior planners (see Section IV-C), by leveraging the VLM for coarse assessments while the LLM handles fine-grained decisions (*e.g.*, floor/region selections), balancing real-time performance and reasoning capability. Key contributions include:

- We introduce an online hierarchical framework for **multi-floor navigation without pre-built maps**, enabling exploration across floors in unseen environments.
- We propose a coarse-to-fine frontier reasoning strategy that **reduces LLM calls by over 90%** compared to prior works, while preserving strong planning performance.
- We demonstrate **state-of-the-art (SOTA) performance** for ZS-OGN benchmarks, improving SR by 7.1% and SPL by 6.8% on HM3D, and SR by 3.4% on MP3D. Beyond simulation, we validate the robustness through **real-world deployment** on a quadruped robot.

II. Related Work

A. Zero-Shot Object-Goal Navigation

ZS-OGN aims to find a target object in an unknown environment without any task-specific training. Early approaches, such as ZSON [8], leveraged VLMs like CLIP [15] to transfer knowledge from Image-Goal to Object-Goal Navigation. Subsequent methods introduced modular designs: ActPept [16] jointly modeled geometric and semantic cues, VLFM [9] introduced vision-language frontier maps, and STRIVE [17] proposed structured representations with VLM-guided navigation. However, most existing methods assumed single-floor environments and lacked explicit multi-floor mechanisms.

B. LLMs for Object-Goal Navigation

Recent ZS-OGN work has turned to LLMs for high-level planning. L3MVN [10] used LLMs to generate long-horizon plans, and SG-Nav [13] constructed scene graphs for contextual memory. InstructNav [12] proposed dynamic chain-of-navigation prompting, and PixNav [11] employed LLMs for room-level exploration. Beyond navigation planning, OpenFMNav [18] and ApexNav [19] leverage LLMs for instruction parsing and detection enhancement respectively. Despite their strong reasoning, frequent LLM API calls cause high computational latency. *ASCENT*addresses this limitation through Coarse-to-Fine Reasoning, significantly reducing LLM invocations while improving planning quality.

C. Multi-Floor Navigation

Navigating across floors poses unique challenges, such as SLAM failures in repetitive or texture-poor environments [3]. Early methods either assumed floor plans or relied on pre-built topological structures [20]. For example, Werby et al. [21] proposed hierarchical scene graphs for language-grounded navigation but required offline map construction. Recently, MFNP [14] pioneered multi-floor ZS-OGN but suffered from spatial map overlap, one-way stair constraints, and heuristic floor switching. We address these through per-floor representations, bidirectional traversal, and hierarchical planning.

III. Methodology

This section introduces our proposed framework *ASCENT*for the OGN task, with an overview shown in Fig. 2. Our approach formulates the OGN problem as minimizing a dual-cost objective that balances exploration and exploitation.

$$\tau = \arg\min \left(\underbrace{\lambda_{\text{expl}} c_{\text{expl}}(\tau)}_{\text{exploration˜cost}} + \underbrace{\lambda_{\text{goal}} c_{\text{goal}}(\tau)}_{\text{exploitation˜cost}} \right) \tag{1}$$

To achieve this dual objective, *ASCENT*incorporates two key components. The **Multi-Floor Abstraction** module minimizes the exploitation cost c_{goal} by ensuring multi-floor accessibility and efficient goal-reaching (see Section III-A). The **Coarse-to-Fine Reasoning** module reduces the exploration cost c_{expl} by prioritizing high-value frontiers (see Section III-B).

A. Multi-Floor Abstraction

To support online navigation in multi-floor environments, our vision-based method eliminates the need for dedicated vertical sensors and abstracts complex 3D environments into a multi-layered representation to facilitate efficient planning. These core innovations through two key designs: **BEV Mapping Representations** and **Multi-Floor Topology Modeling**.

1) BEV Mapping Representations: Our approach builds on the concept of using Bird's Eye View (BEV) representations for frontier-based navigation, inspired by VLFM. Frontiers are defined as the midpoints of the boundaries between explored and unexplored areas, denoted as $\mathcal{F}$. For intra-floor exploration, we use two complementary representations:

Exploration Value Map $\mathcal{M}_{val}$: This map guides efficient local exploration by integrating a *Semantic Similarity Map* $\mathcal{M}_{\text{ss}}$ with a proximity-based *Exploration Cost Map* $\mathcal{M}_{\text{ec}}$. Compared to VLFM, which only use $\mathcal{M}_{\text{ss}}$, it ensures the robot fully exploits local, high-value areas before distant exploration. The total value for the i-th frontier $\mathcal{F}_i$ is defined as:

$$\mathcal{M}_{\text{val}}(\mathcal{F}_i) = \mathcal{M}_{\text{ss}}(\mathcal{F}_i) + \begin{cases} \exp(-d_i) & \text{if } d_i \le d_\theta \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where d_i represents the Euclidean distance from frontier $\mathcal{F}_i$ to the robot's position at the current moment t, d_θ is a distance threshold, and $\mathcal{M}_{\text{ss}}$ aggregates values from $[0, t]$. The selected frontier is converted to a 2D waypoint goal and passed to the Point-Goal Navigation (PointNav) controller, which executes navigation for up to T/10 steps until the frontier is explored or the waypoint becomes unreachable. Intuitively, this design encourages the robot to prioritize frontiers that are both semantically relevant (high $\mathcal{M}_{\text{ss}}$) and spatially accessible (small d_i), ensuring that promising neighbor areas are exploited fully.

Stair-Aware Obstacle Map $\mathcal{M}_{obs}$: This is a binary occupancy grid constructed from depth data, which also aggregates values from $[0, t]$. Unlike VLFM, where obstacle maps treat stairs as impassable, ours re-labels stair-like structures as traversable after stair detection process. The map also maintains persistent blacklist/cache records of failed/successful transitions, providing the topological basis for multi-floor navigation.

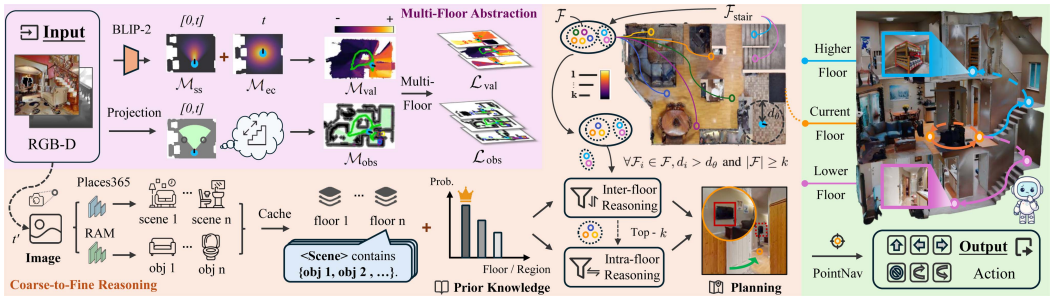


Fig. 2. Framework overview of *ASCENT* . The system takes RGB-D inputs (top-left), and outputs navigation actions (bottom-right). The Multi-Floor Abstraction module (top) builds intra-floor BEV maps and models inter-floor connectivity. The Coarse-to-Fine Reasoning module (bottom) uses the LLM for contextual reasoning across floors. Therefore, *ASCENT*achieves floor-aware, Zero-Shot Object-Goal Navigation.

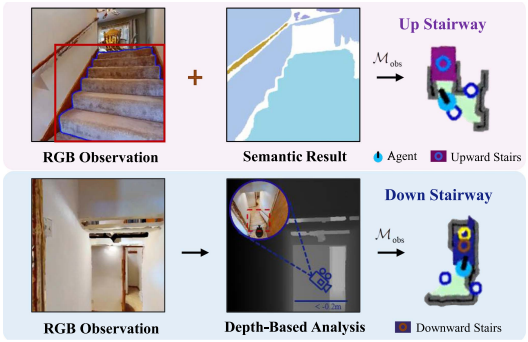


Fig. 3. Stair Detection process. *ASCENT*detects upward stairs (top) and infers downward stairs using depth-based analysis (bottom).

2) Multi-Floor Topology Modeling: To enable online multi-floor navigation, our framework introduces two key designs: a **Stair Detection** module to identify traversable stairs and a **Cross-Floor Transition** module to execute inter-floor movements. Our approach provides superior robustness in both map representation and navigation policy compared to MFNP. **Stair Detection.** Our method detects traversable stairs by combining object detection with semantic segmentation. Let S_{cand} denote the set of staircase bounding boxes detected by the object detector, where each box s has detection confidence $D(s)$. As shown in Fig. 3, a staircase candidate $s \in S_{\text{cand}}$ is considered valid only when $D(s)$ exceeds threshold ϵ_1 and the stair pixel proportion $\frac{\|\mathcal{P}_{\text{stair}}(s)\|}{\|\mathcal{P}_{\text{total}}(s)\|}$ exceeds ϵ_2.

$$S_{\text{valid}} = \left\{ s \in S_{\text{cand}} \,\middle|\, D(s) > \epsilon_1 \wedge \frac{\|\mathcal{P}_{\text{stair}}(s)\|}{\|\mathcal{P}_{\text{total}}(s)\|} > \epsilon_2 \right\} \quad (3)$$

Furthermore, vision-based methods often fail to detect downward stairs in unexplored environments because their pixels are not visible from the current viewpoint, which is also a limitation of methods like MFNP. To address this, we identify areas with depth values less than -0.2 m as potential downward stairs. The robot then executes a LOOK DOWN action and approaches the area to validate the potential stair, enabling robust bidirectional detection for seamless transitions. **Cross-Floor Transition**. When a valid stair region is identified, the robot establishes a specific frontier $\mathcal{F}_{\text{stair}}$ at its midpoint. When the robot makes a cross-floor decision, it first navigates toward $\mathcal{F}_{\text{stair}}$ until reaching the stair area. Then its navigation target switches to a dynamic intermediate waypoint positioned at a distance of $d_{\text{stair}} = 0.8\,\text{m}$ ahead of its current pose. This encourages continuous forward movement while preventing premature stopping or localization drift. The low-level controller leverages a pre-trained PointNav policy.

For unexplored stairs, climbing continues until the robot exits the opposite side. For previously traversed stairs, the robot uses recorded start/end points for direct navigation. If climbing exceeds $\frac{T}{10}$ steps, the robot reverses to its previous position and marks the stair as impassable on $\mathcal{M}_{\text{obs}}$.

Upon floor transition, the robot updates its per-floor BEV map, forming map lists like $\mathcal{L}_{\text{obs}} = [\mathcal{M}_{\text{obs}}^{(1)}, \ldots, \mathcal{M}_{\text{obs}}^{(N)}]$ and $\mathcal{L}_{\text{val}} = [\mathcal{M}_{\text{val}}^{(1)}, \ldots, \mathcal{M}_{\text{val}}^{(N)}]$. Unlike MFNP that merges multi-floor data, our method prevents spatial overlap and inherently handles navigation across multiple floors. When multiple stairs connect the same floors, the robot explores new stairs initially but prioritizes previously traversed stairs or the largest stairs area for subsequent visits, ensuring robust floor revisits.

Overall, we integrate multiple feasibility checks: connectivity validation through $\mathcal{M}_{\text{obs}}$, passability verification via dual-modality stair detection, and access control through black-list/cache mechanisms, ensuring robust multi-floor navigation.

B. Coarse-to-Fine Reasoning

To enable context-aware exploration in multi-floor environments, we introduce a two-stage reasoning pipeline: **Coarse-Grained Assessment** identifies high-value frontiers efficiently, and **Fine-Grained Decision** performs deeper contextual reasoning to select the final target. We leverage statistical priors from the training split of benchmarks to guide exploration, analogous to how learning-based methods capture in-domain knowledge without task-specific training.

1) Coarse-Grained Assessment: For computational efficiency, we first generate a set of distinct frontier proposals by evaluating image-text similarity within $\mathcal{M}_{\text{val}}$. For each detected frontier, at its corresponding moment t', the RGB image is processed to create a structured scene description using a scene classification model and an image tagging model. This description captures both the room type and associated objects, providing rich context. We then rank all available frontiers by their $\mathcal{M}_{\text{val}}$ scores, and the top-k proposals are cached for potential fine-grained analysis.

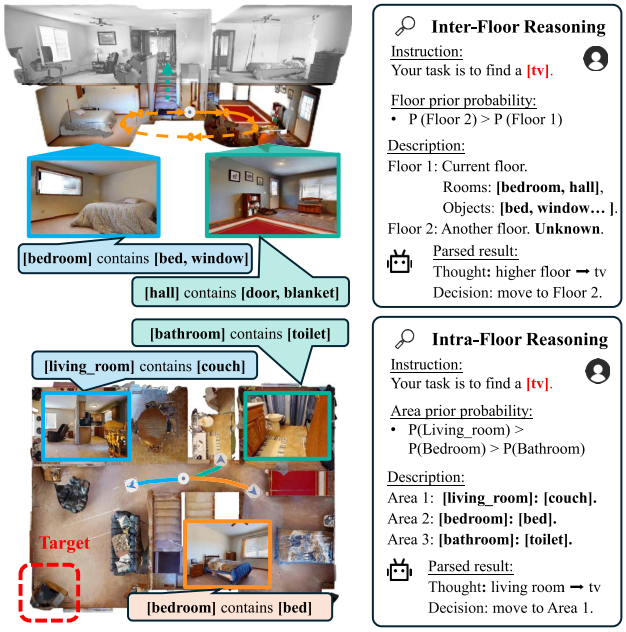


Fig. 4. *Illustration of Fine-Grained Decision:* Following a coarse-grained assessment, the robot feeds cached contextual information and learned object priors to the LLM, which then decides whether to perform inter-floor transition or intra-floor navigation.

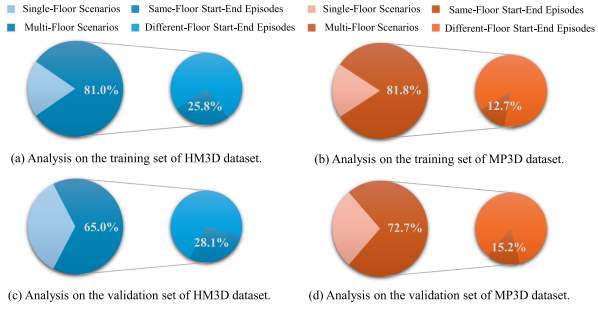


Fig. 5. Multi-floor scenario statistics in OGN benchmarks. Across HM3D and MP3D, over half of scenarios involve multiple floors, with approximately 20% requiring cross-floor navigation.

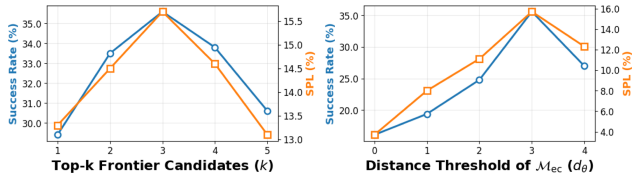


Fig. 6. Hyperparameter sensitivity analysis. Grid search on three scenes of MP3D: optimal performance at $k = 3$ and $d_\theta = 3.0$.

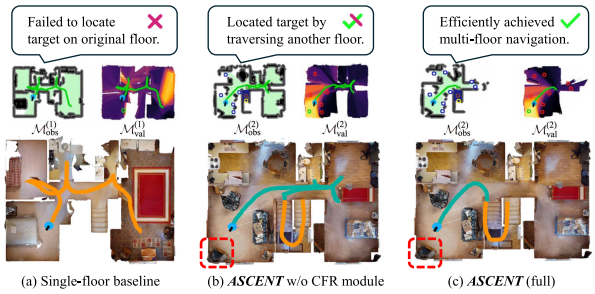


Fig. 7. Qualitative analysis in multi-floor environments. (a) the single-floor baseline, (b) a variant of our method without the Coarse-to-Fine Reasoning (CFR) module, and (c) the complete *ASCENT*.

2) Fine-Grained Decision: By default, the robot directly selects the frontier with the highest score in $\mathcal{M}_{\text{val}}$ for efficiency. This process is activated only when specific conditions are met: $\forall \mathcal{F}_i \in \mathcal{F}, d_i > d_\theta$ and $|\mathcal{F}| \geq k$, where $|\mathcal{F}|$ represents the total number of available frontiers and k is the minimum frontier number threshold. Intuitively, this means LLM-based reasoning is triggered when no frontiers are nearby and sufficient frontier options exist for meaningful comparison. As shown in Fig. 4, the robot performs contextual reasoning using cached information and LLMs in a sequential manner:

Inter-Floor Reasoning: When stair-related frontiers $\mathcal{F}_{\text{stair}}$ exist, the LLM first evaluates whether to switch floors by using contextual information from cached floor descriptions and learned object priors. A policy prevents frequent transitions by requiring an empirically-set minimum time interval $\frac{T}{10}$ steps or full floor coverage. If the decision is to switch floors, the robot proceeds with cross-floor transition.

Intra-Floor Reasoning. If the robot decides to remain on the current floor, the LLM analyzes semantic descriptions of available frontiers and prioritizes the most relevant location based on the task instruction (*e.g.*, "find the cabinet") and frontier context (*e.g.*, room type, object presence).

This hierarchical design balances computational efficiency with contextual relevance, enabling multi-floor spatial reasoning while maintaining real-time performance.

IV. Experiments

A. Experimental Settings

Datasets: We evaluate on HM3D [1] and MP3D [2], the official Habitat Challenge benchmarks for OGN [22]. These benchmarks span 1,000 (HM3D) and 90 (MP3D) realistic scenes across residential and commercial buildings. HM3D contains 2,000 validation episodes across 20 scenes with 6 object categories, while MP3D includes 2,195 validation episodes across 11 scenes with 21 object categories.

As shown in Fig. 5, a significant portion of these scenes are multi-floor: roughly 65% of validation scenarios in HM3D and 73% in MP3D, with about 28% of episodes in HM3D and 15% in MP3D require cross-floor navigation. These statistics establish the critical need for multi-floor navigation. **Metrics.** We adopt two standard metrics from OGN evaluation [23]: Success Rate (SR) and Success weighted by Path Length (SPL) to measure navigation performance and path efficiency. All results are reported in percentages.

Implementation Details: Our experiments are conducted in the Habitat [24]. For object detection, we use D-FINE [25] for COCO-class objects and Grounding-DINO [26] for open-set detection, following VLFM's confidence thresholds. Object segmentation uses Mobile-SAM [27], and stair semantic segmentation uses RedNet [28]. For frontier reasoning, we employ BLIP-2 [29] for semantic similarity, Places365 [30] for scene classification, RAM [31] for object tagging, and Qwen2.5-7B-Instruct [32] for LLM-based reasoning. Our method leverages

TABLE I

COMPARISONS WITH SOTA METHODS. THE TABLE CONTRASTS LEARNING-BASED AND ZERO-SHOT METHODS ON HM3D AND MP3D DATASETS ACROSS THE *"SUCCESS RATE"* (SR) AND *"SUCCESS WEIGHTED BY PATH LENGTH"* (SPL) METRICS. WE INTRODUCE COLUMNS FOR *VISION* AND *LANGUAGE* MODELS TO SPECIFY THE INSTRUCTION INTERPOLATOR COMPONENTS USED BY EACH APPROACH. ALL BASELINE RESULTS FROM THEIR ORIGINAL PUBLICATIONS.

	Method	Venue	Instruction Interpolator		HM3D [1]		MP3D [2]	
			Vision	Language	SR ↑	SPL ↑	SR ↑	SPL ↑
Category: Learning-Based								
Single-Floor	RIM [4]	IROS'23	-	-	57.8	27.2	50.3	17.0
	OVG-Nav [7]	RAL'24	-	-	-	-	35.8	12.3
Multi-Floor	PIRLNav [5]	CVPR'23	-	-	64.1	27.1	-	-
	XGX [6]	ICRA'24	-	-	72.9	35.7	-	-
Category: Zero-Shot								
Single-Floor	ZSON [8]	NeurIPS'22	CLIP [15]	-	25.5	12.6	15.3	4.8
	L3MVN [10]	IROS'23	-	GPT-2 [33]	50.4	23.1	34.9	14.5
	PixNav [11]	ICRA'24	LLaMA-Adapter [34]	GPT-4 [35]	37.9	20.5	-	-
	VLFM [9]	ICRA'24	BLIP-2 [29]	-	52.5	30.4	36.4	17.5
	SG-Nav [13]	NeurIPS'24	LLaVA-1.6-7B [36]	GPT-4 [35]	54.0	24.9	40.2	16.0
	OpenFMNav [18]	NAACL-F'24	GPT-4V [35]	GPT-4 [35]	54.9	24.4	37.2	15.7
	ActPept [16]	RAL'24	GraphSAGE [37]	-	-	-	39.8	17.4
	InstructNav [12]	CoRL'24	GPT-4V [35]	GPT-4 [35]	58.0	20.9	-	-
	ApexNav [19]	RAL'25	BLIP-2 [29]	DeepSeek-V3 [38]	59.6	33.0	39.2	**17.8**
Multi-Floor	MFNP [14]	ICRA'25	Qwen-VLChat-Int4 [39]	Qwen2-7B [40]	58.3	26.7	41.1	15.4
	ASCENT	**Ours**	BLIP-2 [29]	Qwen2.5-7B [32]	**65.4**	**33.5**	**44.5**	15.5

two offline statistical priors: (1) Floor-level priors: distribution of goal objects across floor levels from training episodes of benchmarks. (2) Area-level priors: object-room co-occurrence probabilities from HM3D statistics. All experiments run on two NVIDIA RTX 3090 GPUs.

B. Main Results

Comparisons with SOTAs: As shown in Table I, our method establishes a new SOTA in ZS-OGN on both HM3D and MP3D. The table reports all baseline results from their original publications and details the foundation models that serve as instruction interpolators, processing visual observations and language instructions for navigation decisions.

On HM3D, we achieve 65.4% SR (95% CI: $[63.3\%, 67.5\%]$) and 33.5% SPL (95% CI: $[31.3\%, 35.7\%]$), outperforming 58.3% SR of MFNP (95% CI: $[56.2\%, 60.4\%]$) by +7.1% SR and +6.8% SPL. Bootstrap confidence intervals (10,000 iterations) confirm statistical significance with non-overlapping CIs. On MP3D, we reach 44.5% SR (95% CI: $[42.4\%, 46.6\%]$) and 15.5% SPL, surpassing 41.1% SR of MFNP (95% CI: $[39.0\%, 43.2\%]$) by +3.4% SR. Although CIs show minimal overlap, the consistent improvement across both datasets and controlled comparisons under identical model settings (see Table V) confirm that gains stem from architectural design. Compared to ApexNav, we trade path efficiency (-2.3% SPL) for notably higher success (+5.3% SR), as our cross-floor capability enables navigation in different-floor episodes where single-floor methods tend to fail.

Generalization Performance. Supervised methods like RIM and XGX demonstrate high performance when trained on their corresponding datasets, achieving 50.3% SR on MP3D and 72.9% SR on HM3D, respectively. However, as shown in Table II, these methods exhibit limited cross-dataset generalization. In contrast, *ASCENT*maintains consistent zero-shot performance across both datasets without any task-specific training. This strong cross-dataset consistency makes our framework well-suited for real-world deployment, where training data may not be available for unseen and complex environments.

Performance Analysis Across Floor Scenarios. As shown in Table III, *ASCENT*demonstrates distinct advantages across different floor scenarios. The standard errors (±) are computed via bootstrap estimation with 10,000 iterations. For same-floor start-end episodes, our method achieves 72.6% SR (± 1.1%), outperforming both baselines. The advantages are more pronounced in different-floor episodes, where *ASCENT*achieves 33.3% SR (± 1.6% SE), substantially outperforming VLFM (+32.9%) and MFNP (+19.9%). Comparable standard errors (± 1.1% vs. ± 1.6%) confirm consistent performance across scenarios. Notably, *ASCENT-Ideal* (ground-truth detection) achieves 49.8% SR on different-floor episodes, proving that 16.5% of cross-floor failures stem from perception limitations rather than planning deficiencies.

TABLE II

GENERALIZATION PERFORMANCE: THE SOTA LEARNING-BASED METHODS SHOW POOR TRANSFERABILITY, WHILE OUR ZERO-SHOT METHOD ACHIEVES STRONG CROSS-DATASET GENERALIZATION.

Method	Training Data	HM3D		MP3D	
		SR ↑	SPL ↑	SR ↑	SPL ↑
RIM [4]	MP3D	57.8	27.2	**50.3**	**17.0**
XGX [6]	HM3D	**72.9**	**35.7**	13.6	5.1
ASCENT	–	65.4	33.5	44.5	15.5

TABLE III

PERFORMANCE COMPARISON ACROSS FLOOR SCENARIOS: START-END REFERS TO ROBOT INITIAL POSITION AND TARGET LOCATION. ALL CONDUCTED ON HM3D, AND RESULTS MARKED WITH † FROM AUTHOR CORRESPONDENCE.

Method	Same-Floor Start-End		Different-Floor Start-End		All Episodes	
	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
VLFM*	64.6	37.3	0.4	0.1	52.5	30.4
MFNP†	68.4	30.5	13.4	9.8	58.3	26.7
ASCENT	72.6	37.7	33.3	14.9	65.4	33.5
ASCENT-Ideal	**74.9**	**45.1**	**49.8**	**22.6**	**70.3**	**41.0**

C. Component Analysis

Ablation Study: To evaluate the contribution of each core component, we conduct a comprehensive ablation study on both HM3D and MP3D benchmarks, as shown in Table IV. **(1) Core Components.** The Exploration Cost Map has the most significant impact: its removal leads to performance losses of 9.1%

TABLE IV
ABLATION STUDY ON CORE COMPONENTS. EACH ROW REMOVES ONE CORE COMPONENT FROM THE FULL *ASCENT* MODEL.

Method	HM3D		MP3D	
	SR ↑	SPL ↑	SR ↑	SPL ↑
w/o Exploration Cost Map	56.3	27.7	37.6	13.1
w/o Cross-Floor Transition	56.7	28.6	39.9	**17.9**
w/o Coarse-to-Fine Reasoning	57.7	28.5	40.8	14.2
w/o Prior Knowledge	62.1	30.1	42.5	15.5
w/o Floor-Level Prior Knowledge	62.7	31.4	43.9	16.3
w/o Area-Level Prior Knowledge	63.8	32.9	43.3	15.0
ASCENT	**65.4**	**33.5**	**44.5**	15.5

TABLE V
EFFECT OF LARGE MODEL CONFIGURATIONS. RESULTS MARKED WITH † FROM AUTHOR CORRESPONDENCE.

Method	Interpolator		HM3D		MP3D	
	Vision	Language	SR↑	SPL↑	SR↑	SPL↑
MFNP†	BLIP-2	Qwen2.5-7B	55.8	24.9	38.5	12.4
MFNP	Qwen-VLChat-Int4	Qwen2-7B	58.3	26.7	41.1	15.4
ASCENT	BLIP-2	Qwen2-7B	65.2	33.3	44.1	15.2
	BLIP-2	Qwen2.5-7B	65.4	33.5	44.5	15.5
	Qwen-VLChat-Int4	Qwen2-7B	67.7	34.9	46.5	18.6
	Qwen-VLChat-Int4	Qwen2.5-7B	**67.9**	**35.0**	**46.8**	**18.8**

SR, 5.8% SPL on HM3D, and 6.9% SR, 2.4% SPL on MP3D, as the robot inefficiently traverses frontiers. Removing Cross-Floor Transition impairs multi-floor navigation, resulting in 8.7% SR drops on HM3D and 4.6% SR loss on MP3D. The absence of Coarse-to-Fine Reasoning causes 7.7% SR losses on HM3D and 3.7% SR losses on MP3D, as the robot becomes trapped in local optima. **(2) Impact of Statistical Priors.** Critically, even without any priors, our method achieves 62.1% SR on HM3D and 42.5% SR on MP3D, substantially outperforming MFNP (58.3% SR on HM3D and 41.1% SR on MP3D), demonstrating robust zero-shot generalization. Incorporating priors yields additional gains of +3.3% SR on HM3D and +2.0% SR on MP3D. Floor-level priors contribute more on HM3D (+2.7% SR vs. +1.6% SR), while area-level priors dominate on MP3D (+1.2% SR vs. +0.6% SR). Combining both priors achieves the best SR.

Beyond the core ablation study, we provide a deeper quantitative analysis of our method's performance across different large model configurations and object detector choices.

Effect of Large Model Components: To evaluate robustness to foundation model selection, we test *ASCENT* across four backend configurations and compare against MFNP as shown in Table V. *ASCENT* outperforms MFNP under the same settings, this confirms that our gains stem primarily from architectural innovation, not merely stronger foundation models. Besides, within *ASCENT* variants, VLM choice (± 2.3-2.5% SR) impacts more than LLM selection (± 0.2-0.4% SR), reflecting our design where VLMs handle continuous $\mathcal{M}_{\text{val}}$ generation while LLMs activate only for high-level planning.

Effect of Object Detectors: As shown in Table VI, *ASCENT*'s performance scales with detector quality. We define *ASCENT-Ideal* as a variant with ground-truth object detection, representing the theoretical upper bound under ideal perception. It achieves up to 70.3% SR on HM3D and 58.6% SR on MP3D. The performance gap on MP3D is much larger than that on HM3D, due to broader open-vocabulary challenges on MP3D. Importantly, under identical detection settings,

TABLE VI
EFFECT OF OBJECT DETECTORS: BETTER DETECTORS LEAD TO BETTER PERFORMANCE. G-DINO REPRESENTS GROUNDINGDINO DETECTOR.

Method	Object Detector	HM3D		MP3D	
		SR ↑	SPL ↑	SR ↑	SPL ↑
VLFM	G-DINO + YOLOv7	52.5	30.4	36.4	17.5
	G-DINO + D-FINE	54.1	33.0	37.5	17.8
	Ideal	62.4	40.0	55.3	29.6
ASCENT	G-DINO + YOLOv7	60.9	29.6	42.4	13.8
	G-DINO + D-FINE	65.4	33.5	44.5	15.5
	Ideal	**70.3**	**41.0**	**58.6**	**30.3**

TABLE VII
COMPUTATIONAL EFFICIENCY COMPARISON. RESULTS ARE AVERAGED OVER RANDOMLY SELECTED SCENARIOS ON HM3D. ∗METHODS REIMPLEMENTED WITH QWEN2.5-32B DUE TO GPT-4 API DEPRECATION.

Method	Single-Floor Scenario				Multi-Floor Scenario			
	SR↑	LLM Calls↓	Steps↓	Runtime (s)↓	SR↑	LLM Calls↓	Steps↓	Runtime (s)↓
L3MVN	51.5	35.0	194.1	1047.7	48.5	38.8	216.5	1173.1
PixNav*	48.2	58.1	278.7	276.4	48.1	62.8	287.5	290.1
SG-Nav	54.2	122.3	300.0	728.5	46.5	149.4	250.8	323.6
InstructNav*	54.4	113.6	320.5	458.8	54.3	108.3	446.5	450.1
ASCENT	**57.6**	**2.0**	**171.4**	**190.3**	**64.8**	**2.7**	**153.3**	**181.6**

ASCENT consistently outperforms VLFM (e.g., G-DINO + YOLOv7, by +8.4% SR on HM3D and +6.0% SR on MP3D), demonstrating that its gains stem from architectural design rather than perception advantages.

D. Efficiency & Sensitivity Analysis

Computational Efficiency Comparison: To evaluate deployment feasibility, we compare computational efficiency with recent LLM-based planners on randomly selected HM3D scenarios (3 single-floor and 3 multi-floor). MFNP is excluded as it is not open source. Due to GPT-4 API deprecation, we re-implemented methods PixNav and InstructNav with Qwen2.5-32B-Instruct for efficiency analysis. As shown in Table VII, *ASCENT* reduces LLM calls by over 90% (2.0-2.7 vs. 35.0-149.4 per episode) while achieving higher SR, better SPL, and shorter runtimes. This efficiency stems from invoking LLMs only for high-level reasoning rather than step-by-step planning.

Hyperparameter Sensitivity Analysis: We conduct a systematic sensitivity analysis for key hyperparameters in our framework. As shown in Fig. 6, we evaluate the impact of top-k frontier candidates (k) and distance threshold of Exploration Cost Map $\mathcal{M}_{\text{ec}}$ (d_θ) through grid search on three MP3D validation scenes. The parameter k controls the number of frontier candidates in Coarse-to-Fine Reasoning, while d_θ determines the spatial range in $\mathcal{M}_{\text{val}}$ for Multi-Floor Abstraction. Results reveal that $k = 3$ and $d_\theta = 3.0$ yield optimal performance, which we thus adopt for all other experiments.

E. Qualitative Analysis

We provide a qualitative analysis to visually show how *ASCENT* minimizes the dual objective of c_{expl} and c_{goal}.

Fig. 7 illustrates the navigation behavior of different methods in a multi-floor scenario. The scenario involves a target object located on the second floor, with the agent starting on the first floor. Panel (a) shows a single-floor baseline that fails to locate the stairway, resulting in mission failure. Panel (b) illustrates a variant of our method without the Coarse-to-Fine Reasoning module. While it successfully navigates to the second floor and reaches the target, it takes an inefficient path. This highlights the

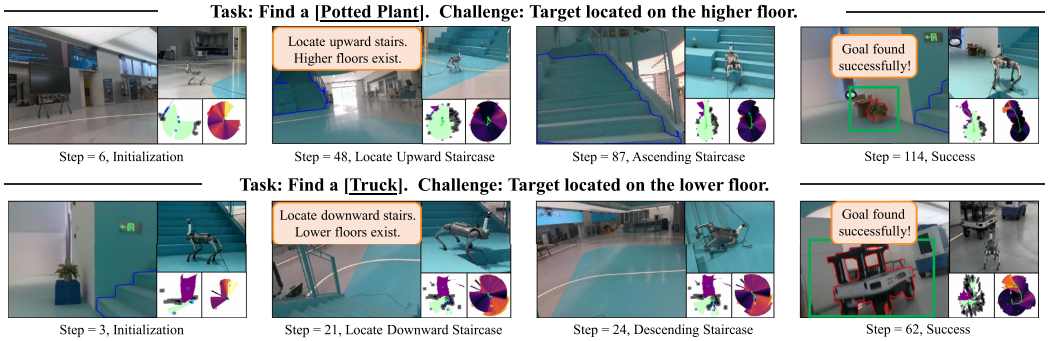
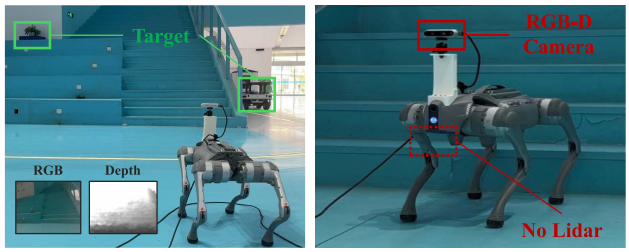


Fig. 8. Real-world deployment for *ASCENT*. Our experiments conduct on the Unitree quadruped robot *Go2*. The top row shows a ascent task to find a "potted plant" on a higher floor. The bottom row illustrates a descent task to find a "truck" on a lower floor.

TABLE VIII
REAL-WORLD EXPERIMENTAL RESULTS. VALUES ARE AVERAGED OVER ALL TRIALS. TD: TRAVELED DISTANCE (M); TT: TRAVELED TIME (S).

Scenario	TD	TT	SR↑	SPL↑
Current-Floor	8.3	75.7	62.5	35.9
Higher-Floor	16.9	200.8	37.5	20.2
Lower-Floor	20.5	264.1	25.0	12.4
Total	15.2	180.2	41.7	22.8



(a) Scene Overview (b) Sensor Configuration

Fig. 9. Real-world experimental setup. (a) Target objects located on different floors and RGB-D egocentric observations from the robot. (b) Our quadruped robot equipped with vision-only sensors.

importance of our high-level reasoning for reducing c_{goal}. Panel (c) shows that our *ASCENT*method not only supports multi-floor navigation but also selects a nearly optimal path to the target upon the second floor, achieving high SR while reducing c_{expl} with superior path efficiency.

F. Real-World Deployment

We deploy *ASCENT*on a Unitree Go2 quadruped robot to validate its effectiveness in real-world multi-floor scenarios. As shown in Fig. 8, the robot successfully navigates multi-floor environments, ascending stairs to find a "potted plant" and descending to locate a "truck", validating our method's generalization from simulation to reality.

We evaluate performance over eight independent trials per scenario (current-floor, higher-floor, and lower-floor), as shown in Table VIII. Results highlight the increasing difficulty of cross-floor navigation: stair ascents challenge perception and locomotion, while descents are further complicated by depth sensing and balance maintenance. Despite these challenges, *ASCENT*achieves consistent performance and efficiency, underscoring its robustness in real-world environments.

The experimental setup is shown in Fig. 9. The robot is equipped with an Intel RealSense D435i RGB-D camera. High-level planning and reasoning are executed externally: the core algorithm runs on a laptop with an RTX 2060 GPU connected via Ethernet, while LLM queries are offloaded to a remote server with dual RTX 3090 GPUs over a wireless link. Low-level locomotion is controlled by a pre-trained PointNav policy, as in simulation, ensuring stable execution while *ASCENT*focuses on high-level decision-making.

G. Robustness Analysis

Recovery Analysis: To evaluate multi-floor robustness, we analyze failure patterns across cross-floor episodes on 3 HM3D scenarios, as shown in Table IX. There are 47.0% cross-floor episodes succeed without recovery, validating our feasibility checks. For failures, we implement two recovery strategies: Replan recovery handles transient issues (detection failure: 33.3%, stuck: 3.4%) via local corrections (12-35 steps), while Backtrack recovery addresses fundamental errors (misidentification: 4.3%, wrong floor: 12.0%) through global retrying (43-198 steps). This dual recovery mechanism maintains robustness across diverse failure cross-floor cases. **Stress Test.** To evaluate robustness under extreme conditions, we identified two HM3D scenes with naturally occurring navmesh corruption where visual appearance suggests traversability but geometric damage creates impassable regions. As shown in Fig. 10, for the cross-floor episodes of these scenes, Scene 1 (severe damage) achieves only 12.5% SR as extensive corruption blocks most traversable paths. Scene 2 (partial corruption) achieves 36.0% SR as partial damage allows exploring alternative trajectories. All failures occur at stair traversal, indicating our system is vulnerable when visual perception contradicts geometric reality.

TABLE IX
RECOVERY ANALYSIS FOR CROSS-FLOOR EPISODES. RECOVERY RATE IS THE PERCENTAGE OF EPISODES WITH SUCCESSFUL FAILURE RECOVERY. RECOVERY COST DENOTES EXTRA TIMESTEPS FROM RECOVERY OPERATIONS.

Recovery Type	Failure Type	Percentage (%)	Recovery Rate (%)	Avg. Cost (steps)
No Recovery	Success (no failure)	47.0	-	-
Backtrack	Stair misidentification	4.3	20.0	43
	Incorrect floor transition	12.0	14.3	198
Replan	Stuck on stair	3.4	75.0	35
	Stair detection failure	33.3	23.1	12

(a) Scene 1: Severe Damage (b) Scene 2: Partial Corruption

Fig. 10. Stress test on visual-geometric mismatch. We test on two HM3D scenes with corrupted stair geometry where appearance suggests traversability but navmesh damage creates impassable regions.

V. Limitations

While OGN assumes static environments, during real-world deployment we have observed that moving obstacles like pedestrians can compromise obstacle mapping accuracy and navigation performance. Future work will integrate adaptive replanning modules for dynamic scenarios. Additionally, our method is designed for normal staircases and may not perform well on spiral or irregular stairs. Finally, while HM3D and MP3D are widely-used multi-floor benchmarks, future work could explore larger-scale buildings or scenarios.

VI. Conclusion

We present **ASCENT**, a floor-aware ZS-OGN framework that addresses the limitations of existing methods in online multi-floor navigation. Our approach enables multi-floor planning and context-aware exploration without requiring task-specific training. Experimental results on HM3D and MP3D benchmarks show that our method outperforms SOTA zero-shot methods. We further validated its real-world applicability through deployment on a quadruped robot. This work offers a training-free solution for online multi-floor navigation.

References

[1] S. K. Ramakrishnan et al., "Habitat-MatterPort 3D dataset (HM3D): 1000 large-scale 3D environments for embodied AI," 2021, *arXiv:2109.08238*.

[2] A. Chang et al., "MatterPort3D: Learning from RGB-D data in indoor environments," in *Proc. Int. Conf. 3D Vis. (3DV)*, 2017, pp. 667–676.

[3] D. Chung and J. Kim, "NV-LIOM: LiDAR-inertial odometry and mapping using normal vectors towards robust SLAM in multifloor environments," *IEEE Robot. Automat. Lett.*, vol. 9, no. 11, pp. 9375–9382, Nov. 2024.

[4] S. Chen, T. Chabal, I. Laptev, and C. Schmid, "Object goal navigation with recursive implicit maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 7089–7096.

[5] R. Ramrakhya, D. Batra, E. Wijmans, and A. Das, "PIRLNav: Pretraining with imitation and RL finetuning for objectNAV," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 17896–17906.

[6] J. Wasserman, G. Chowdhary, A. Gupta, and U. Jain, "Exploitation-guided exploration for semantic embodied navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 2901–2908.

[7] H. Yoo, Y. Choi, J. Park, and S. Oh, "Commonsense-aware object value graph for object goal navigation," *IEEE Robot. Automat. Lett.*, vol. 9, no. 5, pp. 4423–4430, May 2024.

[8] 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.

[9] 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.*, 2024, pp. 42–48.

[10] 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.*, 2023, pp. 3554–3560.

[11] W. Cai et al., "Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 5228–5234.

[12] 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.

[13] H. Yin, X. Xu, Z. Wu, J. Zhou, and J. Lu, "SG-Nav: Online 3D scene graph prompting for LLM-based zero-shot object navigation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, vol. 37, pp. 5285–5307.

[14] L. Zhang et al., "Multi-floor zero-shot object navigation policy," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2025, pp. 6416–6422.

[15] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[16] Y. Guo et al., "An object-driven navigation strategy based on active perception and semantic association," *IEEE Robot. Automat. Lett.*, vol. 9, no. 8, pp. 7110–7117, Aug. 2024.

[17] H. Zhu et al., "STRIVE: Structured representation integrating VLM reasoning for efficient object navigation," 2025, *arXiv:2505.06729*.

[18] Y. Kuang, H. Lin, and M. Jiang, "OpenFMNav: Towards open-set zero-shot object navigation via vision-language foundation models," 2024, *arXiv:2402.10670*.

[19] M. Zhang et al., "ApexNav: An adaptive exploration strategy for zero-shot object navigation with target-centric semantic fusion," *IEEE Robot. Automat. Lett.*, vol. 10, no. 11, pp. 11530–11537, Nov. 2025, doi: 10.1109/LRA.2025.3606388.

[20] T. Kim, G. Kang, D. Lee, and D. H. Shim, "Development of an indoor delivery mobile robot for a multi-floor environment," *IEEE Access*, vol. 12, pp. 45202–45215, 2024.

[21] A. Werby, "Hierarchical open-vocabulary 3D scene graphs for language-grounded robot navigation," in *Proc. Robot.: Sci. Syst.*, 2024, Art. no. 077.

[22] K. Yadav et al., "Habitat challenge 2022," 2022. [Online]. Available: https://aihabitat.org/challenge/2022/

[23] D. Batra et al., "ObjectNav revisited: On evaluation of embodied agents navigating to objects," 2020, *arXiv:2006.13171*.

[24] M. Savva et al., "Habitat: A platform for embodied AI research," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9339–9347.

[25] Y. Peng, H. Li, P. Wu, Y. Zhang, X. Sun, and F. Wu, "D-Fine: Redefine regression task in detrs as fine-grained distribution refinement," 2024, *arXiv:2410.13842*.

[26] S. Liu et al., "Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2024, pp. 38–55.

[27] C. Zhang et al., "Faster segment anything: Towards lightweight sam for mobile applications," 2023, *arXiv:2306.14289*.

[28] J. Jiang, L. Zheng, F. Luo, and Z. Zhang, "RedNet: Residual encoder-decoder network for indoor RGB-D semantic segmentation," 2018, *arXiv:1806.01054*.

[29] 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.

[30] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018.

[31] Y. Zhang et al., "Recognize anything: A strong image tagging model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 1724–1732.

[32] A. Yang et al., "Qwen2.5 technical report," 2024, *arXiv:2412.15115*.

[33] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[34] R. Zhang et al., "LLaMA-Adapter: Efficient fine-tuning of language models with zero-init attention," 2023, *arXiv:2303.16199*.

[35] J. Achiam et al., "GPT-4 technical report," 2023, *arXiv:2303.08774*.

[36] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, vol. 36, pp. 34892–34916.

[37] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1024–1034.

[38] A. Liu et al., "DeepSeek-V3 technical report," 2024, *arXiv:2412.19437*.

[39] J. Bai et al., "Qwen-Vl: A versatile vision-language model for understanding, localization," 2023, *arXiv:2308.12966*.

[40] Q. Team, "Qwen2 technical report," 2024, *arXiv:2407.10671*.