Bridging the 2D-3D Gap: A Hierarchical Semantic-Geometric Map for Vision Language Navigation

Supplementary Material

In this supplementary material, we provide additional details and comprehensive analyses to complement the main paper. Sec. 1 elaborates on the specific implementation details, covering the mechanism for multi-floor navigation, simulator configurations, and algorithmic hyperparameters. Sec. 2 presents extended quantitative results, including the performance evaluation on the Object Goal Navigation task and a controlled comparative analysis to disentangle the impact of the VLM backbone from our framework design. Finally, Sec. 3 showcases more qualitative results, offering detailed visualizations of successful navigation trajectories and an in-depth analysis of typical failure cases.

1. More Implementation Details

1.1. Implementation of Multi-Floor Navigation

To enable robust navigation in complex, multi-stair environments, we extend the HSGM framework to manage independent map representations for each floor. Let $\mathcal{F} = \{f_0, \ldots, f_n\}$ denote the set of floors. For each floor f_i, the system maintains a dedicated map instance $\mathcal{M}^{(i)}$ containing the floor-specific scene point cloud $\mathcal{P}^{(i)}_{scene}$, navigable areas $\mathcal{P}^{(i)}_{nav}$, and obstacles $\mathcal{P}^{(i)}_{obs}$. The active floor map is dynamically updated according to the following mechanism.

Floor Transition Mechanism. Floor transitions are governed by a state machine that monitors the agent's height relative to the current floor. A floor switch is triggered only when two conditions are satisfied: (1) the agent is located on a valid *platform* (e.g., a landing), and (2) a significant vertical displacement is detected. Specifically, platforms are detected using RANSAC plane fitting on local points within a radius of $r = 1.5$m. A candidate plane Π is validated if it exhibits a near-vertical normal ($|n_z| \geq 0.95$) and sufficient support area. Let $\Delta H = z_{curr} - h^{last}_{floor}$ denote the vertical displacement relative to the previous floor height. The floor switching condition is then formulated as:

$$\text{Switch} = \mathbb{I}(\text{isPlatform}(\Pi_{curr})) \wedge (|\Delta H| \geq \alpha \cdot H_{floor}), \tag{1}$$

where H_{floor} is the estimated floor-to-ceiling height and $\alpha = 0.75$ is a threshold factor. Crucially, this transition triggers a synchronous update of the visual input: the 2D BEV map $\mathcal{M}_{\text{bev}}$ is immediately switched to the rasterized representation of the new floor $\mathcal{M}^{(f_{\text{new}})}$, ensuring the VLM perceives the correct spatial context for subsequent planning.

Table 1. **Performance comparison on Object Goal Navigation. Bold** denotes the best zero-shot result, and underline denotes the second best.

Method	Zero-shot	SR ↑	SPL ↑
Navid [7]	×	32.5	21.5
MapNav [8]	×	34.6	25.6
Uni-Navid [6]	×	73.7	37.1
vlfm [5]	✓	63.6	32.5
PIVOT [3]	✓	24.6	10.6
InstructNav [2]	✓	58.0	20.9
ApexNav [9]	✓	**76.2**	**38.0**
HSGM (Ours)	✓	<u>73.6</u>	<u>36.3</u>

Staircase Modeling and Planning. Stairs are geometrically distinct from flat terrain, often resembling obstacles due to their slope. To facilitate traversal, we explicitly identify stair regions $\mathcal{P}_{stair}$ by filtering scene points based on surface normal inclination, retaining points where the vertical component $|n_z| \in [0.2, 0.7]$. These points are spatially clustered using DBSCAN to filter noise. During low-level planning, a waypoint $\mathbf{w}$ with a cylindrical agent footprint $\mathcal{C}(\mathbf{w})$ is considered valid if it is either collision-free or located within a detected stair region:

$$\text{isValid}(\mathbf{w}) = (\mathcal{C}(\mathbf{w}) \cap \mathcal{P}_{obs} = \emptyset) \vee (\mathcal{C}(\mathbf{w}) \cap \mathcal{P}_{stair} \neq \emptyset). \tag{2}$$

This mechanism effectively exempts staircases from standard obstacle constraints, allowing the A^* planner to generate continuous paths across different elevations while maintaining safety on flat ground.

1.2. Additional Experimental Settings

Simulator Configuration. We conduct our experiments using the Habitat simulator. The agent is modeled with a physical height of 1.2 m. The onboard visual sensor is configured with a Horizontal Field of View (HFOV) of 135° and a downward tilt angle (pitch) of 35° to optimize ground visibility. All visual observations are rendered at a resolution of 480×640 pixels.

Algorithmic Hyperparameters. In the waypoint generation phase, we employ a cylindrical collision model for the agent with a height of 1.2 m and a radius of 0.2 m to ensure geometric feasibility. For the low-level motion controller, the A^* path planning algorithm is constrained to a maxi-

Table 2. **Comprehensive Analysis of Backbone vs. Framework.** We compare AO-Planner and HSGM under different configurations. Even without the BEV map, our decoupled framework (using only egocentric views) drastically outperforms the GPT-5-powered AO-Planner, highlighting the superiority of our navigation paradigm.

Method	Input View	Backbone	SR ↑	SPL ↑	NE ↓	OSR ↑
AO-Planner [1]	Ego-only	GPT-4o + Gemini 1.5pro	25.5	16.6	6.95	38.3
AO-Planner†	Ego-only	**GPT-5**	32.3	21.3	6.12	45.0
HSGM (w/o BEV)	Ego-only	**GPT-5**	46.0	30.1	5.37	61.7
HSGM (Full)	**Ego + BEV**	**GPT-5**	**51.0**	**33.7**	**5.24**	**61.7**
HSGM (Full)	**Ego + BEV**	**GPT-4o**	41.3	30.1	6.76	51.0

†: Re-implemented using the GPT-5 API.

mum of 100 iterations. The maximum trajectory length is capped at 200 steps per episode.

Hardware Specifications. All experiments were executed on a workstation equipped with 96 GB of RAM and a single NVIDIA GeForce RTX 4090 GPU.

2. Supplementary Results

2.1. Performance on Object Goal Navigation

To further assess the versatility and generalization capability of our framework beyond instruction-following, we evaluated HSGM on the Object Goal Navigation (ObjectNav) task using the challenging Habitat-Matterport 3D (HM3D) [4] dataset. In this setting, the complex narrative instructions are replaced with a standardized template: "*Navigate to the target object [object] and get as close to it as possible.*" Following standard protocols, a navigation episode is considered successful if the agent stops within a Euclidean distance of **0.3m** from the target object.

We compare HSGM against several supervised and zero-shot navigation agents. The results are summarized in Table 1.

Results and Analysis. Our framework achieves a remarkable Success Rate (SR) of **73.6%** and an SPL of **36.3%**. First, compared to other zero-shot baselines, HSGM significantly outperforms methods like PIVOT (24.6% SR), InstructNav (58.0% SR), and vlfm (63.6% SR). This demonstrates that our hierarchical semantic-geometric map provides a more robust representation for object localization and path planning than pure frontier-based or heuristic approaches. Second, while slightly outperformed by ApexNav (76.2% SR), which is specifically optimized for this exploration task, HSGM remains highly competitive, securing the second-best performance among all zero-shot methods. Most notably, our zero-shot approach performs on par with the best supervised method, Uni-Navid (73.7% SR), and substantially surpasses earlier supervised methods like Navid (32.5% SR) and MapNav (34.6% SR). This result validates that our explicit 3D mapping and decoupled planning strategy can achieve human-level perception and planning capabilities on the HM3D scenes without requiring extensive domain-specific training data.

2.2. Disentangling Model Capabilities from Methodological Contributions

A critical question in evaluating zero-shot VLN methods is determining how much performance gain originates from the foundation model (e.g., GPT-5) versus the navigation framework itself. To provide a comprehensive answer, we conducted a controlled comparative analysis involving four settings, as detailed in Table 2.

Baselines and Variants. We compare the following configurations:

- **AO-Planner (Original)**: The reported performance of the baseline using its default VLM [1].
- **AO-Planner (GPT-5)**: Our re-implementation of AO-Planner using the exact same GPT-5 API as our method, evaluated on the 300-episode subset.
- **HSGM (w/o BEV Map)**: Our ablation variant where the VLM relies solely on egocentric visual prompts (similar to AO-Planner's input) without the top-down BEV map representation.
- **HSGM (Full Model)**: Our complete framework incorporating the Hierarchical Semantic-Geometric Map. We evaluate this setting using both **GPT-5** and **GPT-4o** backbones.

Results and Analysis. The results indicate that while upgrading AO-Planner to GPT-5 improves its Success Rate to 32.3%, it still significantly lags behind our method. The most critical comparison lies between **AO-Planner (GPT-5)** and our variant **HSGM (w/o BEV Map)**. Despite both utilizing the same backbone and egocentric inputs, our decoupled framework achieves a Success Rate of **46.0%**, outperforming the upgraded AO-Planner by a substantial margin of **13.7%**. This disparity exposes fundamental flaws

Table 3. **Latency Analysis. Top:** ID: Instr. Decompose, IS: Inst. Seg., PU: PCD Update, WG: Waypoint Gen., BR: BEV Raster., VQ: VLM Query (GPT-5), PP: Path Plan.

Module	**ID**	**IS**	**PU**	**WG**	**BR**	**VQ**	**PP**
Latency	3.1s	27ms	165ms	130ms	77ms	23.8s	11ms

Settings	**HSGM (Ours)**				**AO-Planner [1]**	
	Dec.	Follow	Step Avg.	Episode	Step Avg.	Episode
Latency	24.3s	192ms	**4.87s**	**341s**	8.68s	895s

in the visual prompting paradigm. First, regarding *perception reliability*, we observed that AO-Planner's 2D segmentation (Grounded SAM) often hallucinates navigable areas on vertical surfaces (e.g., walls) due to texture similarities, causing the VLM to plan collision-prone paths. In contrast, HSGM employs 3D geometric constraints derived from depth data to physically enforce obstacle avoidance. Second, concerning the *planning domain*, AO-Planner forces the VLM to infer 3D spatial dynamics implicitly from 2D pixels. Our method resolves this by fully decoupling reasoning from execution: the VLM solely identifies high-level waypoints, while the robust A^* algorithm ensures precise low-level control. Thirdly, incorporating the global BEV Map in our Full Model further extends the lead to **51.0%** SR, confirming the additional value of explicit global spatial modeling. Finally, when substituting the backbone with a less capable model(GPT-4o), HSGM still achieves a **41.3%** SR.

2.3. Latency and Token Cost

We analyze system latency and VLM token consumption to evaluate efficiency. Agent operations are divided into *Decision steps* (all modules invoked) and *Path Following steps* (only perception and map updates). As shown in Table 3, while Decision steps take 24.3s (dominated by the 23.8s VLM query), Path Following steps are highly efficient (192ms). Since multiple fast following steps occur between decisions, the overall average step latency is significantly amortized to **4.87s**. Compared to AO-Planner [1], HSGM achieves superior navigation performance with nearly half the average step latency (4.87s vs. 8.68s) and a drastically shorter total episode time (**341s** vs. 895s). In terms of token consumption, HSGM maintains a moderate level, with 319 tokens per decision and 4,229 tokens per episode.

3. More Visualization Results

3.1. Success Cases

As illustrated in Figure 1, our framework demonstrates robust performance across diverse and complex indoor environments. The visualization underscores the system's ability to decompose complex natural language instructions into manageable sequential subtasks, providing a clear and structured roadmap for long-horizon navigation. By synergizing global context from the BEV Map with local waypoint visual prompts, the VLM acts as an intuitive high-level planner, making reliable decisions to select geometrically valid targets.

Critically, our decoupled architecture ensures that these high-level semantic decisions are translated into precise physical actions. Once a target waypoint is selected, the underlying algorithm plans an optimal, collision-free path using A^*, enabling the agent to safely traverse cluttered environments that typically challenge end-to-end models. Furthermore, upon reaching the designated location for a specific subtask, the VLM effectively verifies the completion status, ensuring smooth transitions between subtasks or precise termination of the episode.

3.2. Failure Case Analysis

Despite achieving state-of-the-art zero-shot performance, our qualitative analysis reveals specific limitations in the spatial reasoning capabilities of current VLMs, particularly regarding global scene understanding and precise self-localization.

The first failure mode, as shown in Figure 2, involves errors in identifying sequential landmarks due to fragmented global perception. In this episode, the agent is instructed to enter the "second door on the left." However, at Step 5, the VLM fails to correctly interpret the global scene structure across multiple egocentric images. Instead of identifying the correct target (Waypoint 1), it incorrectly identifies the door corresponding to Waypoint 4 as the target. This misidentification highlights the VLM's limited capability in stitching together temporal observations to form a coherent global scene understanding, leading to failures in tasks requiring sequential counting.

The second failure mode, illustrated in Figure 3, pertains to the premature execution of actions caused by inaccurate state estimation. The instruction explicitly requires the agent to move along the hallway until reaching the end (Waypoint 3) before turning. However, at Step 2, due to an inaccurate understanding of its own position relative to the corridor's geometry, the VLM incorrectly determines that it has already reached the end of the hallway. Consequently, it prematurely executes a left turn (L) at an intermediate junction. This failure suggests that while the HSGM provides geometric layout, the VLM occasionally struggles to ground strict locational constraints (e.g., "at the end") against its current spatial state, prioritizing immediate directional affordances over geometric termination conditions.

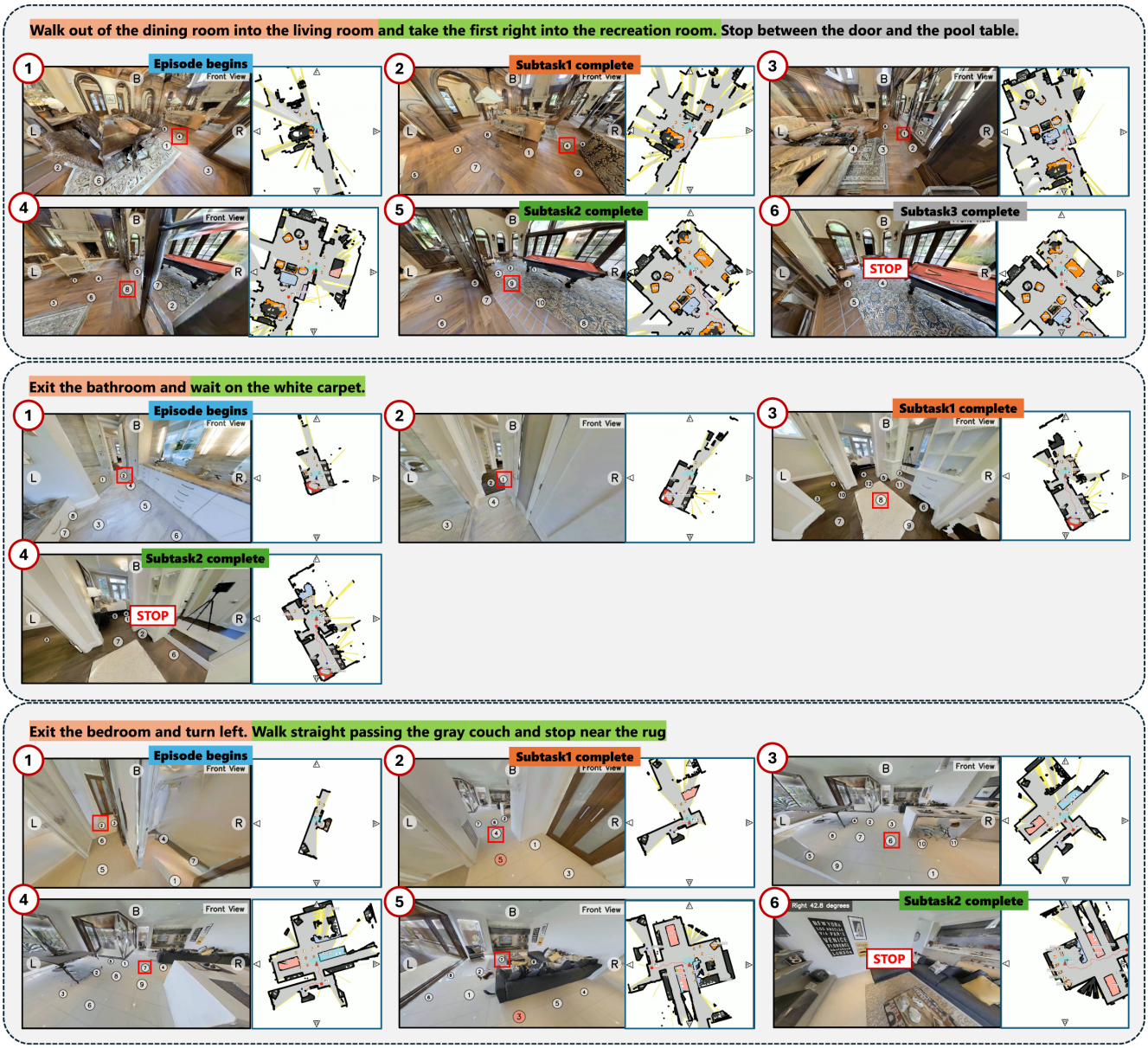

Figure 1. **Visualization of Success Cases.** We showcase three episodes demonstrating the agent's capability in multi-room traversal, object-referenced navigation (e.g., "passing the gray couch"), and precise destination identification (e.g., "wait on the white carpet").

References

[1] Jiaqi Chen, Bingqian Lin, Xinmin Liu, Lin Ma, Xiaodan Liang, and Kwan-Yee K Wong. Affordances-oriented planning using foundation models for continuous vision-language navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 23568–23576, 2025. 2, 3

[2] Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. *arXiv preprint arXiv:2406.04882*, 2024. 1

[3] Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, et al. Pivot: iterative visual prompting elicits actionable knowledge for vlms. In *Proceedings of the 41st International Conference on Machine Learning*, pages 37321–37341, 2024. 1

[4] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2

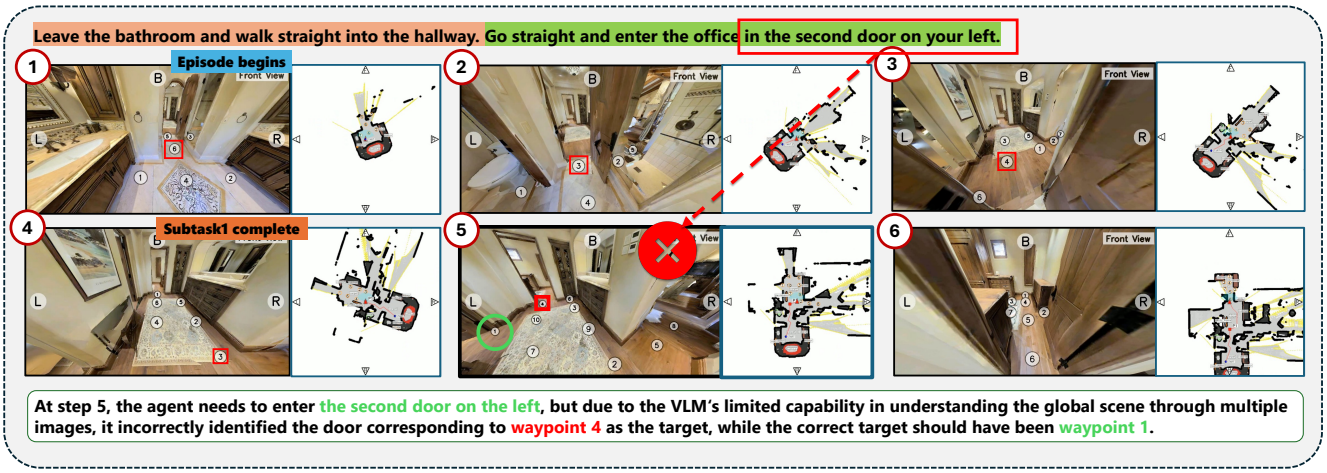

Figure 2. **Failure Case 1: Sequential Counting Error.**

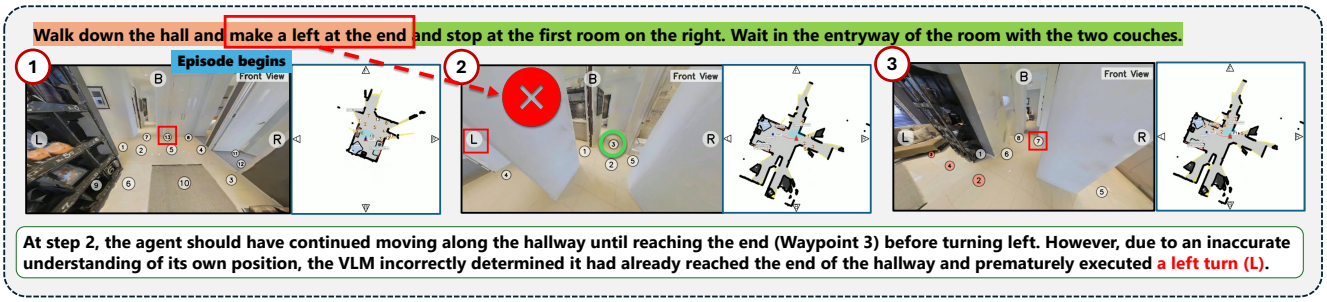

Figure 3. **Failure Case 2: Premature Execution.**

[5] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 42–48. IEEE, 2024. 1

[6] Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *arXiv preprint arXiv:2412.06224*, 2024. 1

[7] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852*, 2024. 1

[8] Lingfeng Zhang, Xiaoshuai Hao, Qinwen Xu, Qiang Zhang, Xinyao Zhang, Pengwei Wang, Jing Zhang, Zhongyuan Wang, Shanghang Zhang, and Renjing Xu. MapNav: A novel memory representation via annotated semantic maps for VLM-based vision-and-language navigation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13032–13056, Vienna, Austria, 2025. Association for Computational Linguistics. 1

[9] Mingjie Zhang, Yuheng Du, Chengkai Wu, Jinni Zhou, Zhenchao Qi, Jun Ma, and Boyu Zhou. Apexnav: An adaptive exploration strategy for zero-shot object navigation with target-centric semantic fusion. *IEEE Robotics and Automation Letters*, 2025. 1