Hi-Nav: Hierarchical Framework for Continuous Vision-Language Navigation via Map Guidance and Waypoint Reasoning

Zhiyu Zhou[1,2,*], Bin Guan[1,*], Wenbin Yang[1,2,*], Zhi Gao[1,2,†], and Hao Fang[3]

[1] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China gaozhinus@gmail.com
[2] Hubei International Science and Technology Cooperation Base for High-End Equipment (Joint Research), Wuhan, China
[3] School of Automation, Beijing Institute of Technology, Beijing, China

Abstract. Despite remarkable advancements in Vision-and-Language Navigation (VLN), a significant sim-to-real gap remains. Most methods are trained and evaluated in simplified or physics-limited simulators, which ignore the feedback loop between decision making and physical execution, resulting in unstable navigation performance. To bridge this gap, we propose Hi-Nav, a top-down hierarchical navigation framework that decomposes VLN into three controllable levels. The high-level module leverages a large language model to split long-horizon instructions into sub-tasks. At the middle level, an Interest Score Occupancy Map (ISOM) integrates instruction-relevant semantics with geometric traversability to perceive the surrounding environment, after which waypoint reasoning is performed for route refinement. ISOM construction and waypoint reasoning are jointly modeled within a unified multi-task learning framework, where a Q-Former fuses RGB-D observations and navigation instructions into shared query representations for efficient multi-task learning, augmented with learnable temporal embeddings to mitigate long-horizon forgetting. Finally, low-level behaviors are executed through ROS, enabling obstacle avoidance. We introduce a novel waypoint reasoning dataset for comprehensive training and build a ROS/Gazebo VLN benchmark to evaluate methods in a realistic simulation setting, which significantly reduces the difficulty of sim-to-real transfer. Extensive experiments conducted in both Gazebo and the real world demonstrate strong performance, achieving up to 80% Success Rate (SR) with 0.65 m Navigation Error (NE). Our code will be released at https://hiway-page.github.io/.

Keywords: Vision-and-Language Navigation · Hierarchical Navigation

1 Introduction

Vision-and-Language Navigation (VLN) studies how an embodied agent comprehends natural-language instructions to navigate to a target location in previously

* Equal contribution.
† Corresponding author.

unseen environments. As a fundamental task, VLN is essential for developing Embodied AI. Early VLN benchmarks are largely built on discretized simulators, where the agent moves by traversing a topology of panoramic images (called a nav-graph) [2, 4, 9, 20, 23, 31]. This graph-based abstraction enables efficient training and evaluation, yet it assumes perfect transitions and localization between graph nodes, limiting generalization and hindering direct deployment in real-world robotic systems.

To mitigate this over-simplified abstraction, VLN in continuous environments (VLN-CE) reformulates navigation as low-level action execution in free space [17,19]. Existing VLN-CE methods typically follow two paradigms: end-to-end policies that directly predict low-level actions from language and egocentric observations [8, 26, 39, 40, 43]; waypoint-based approaches that predict navigable intermediate targets executed by a local controller [7,11,24,33,37,41]. Meanwhile, large-scale vision–language models and the growth of VLN training data have driven rapid progress in VLN-CE performance [34, 39, 40]. However, sim-to-real transfer remains challenging: policies trained in Habitat-style simulators often experience severe degradation when transferred to real robots [18, 25, 29, 30], and real-robot studies further highlight the impact of imperfect localization and execution errors [1, 27]. These observations suggest that robust VLN requires not only stronger vision–language grounding, but also a controllable structure that can cope with compounding and robotics execution constraints.

To bridge this gap, we propose Hi-Nav, a hierarchical framework that structures VLN-CE into three levels. The high-level uses an LLM to split long-horizon instructions into semantic sub-tasks and track stage progress. The middle level builds an Interest Score Occupancy Map (ISOM) that combines instruction-relevant semantics with geometric traversability to perceive the environment, and then refines the route through waypoint reasoning. Both ISOM construction and waypoint reasoning are learned jointly in a unified multi-task model: a Q-Former fuses RGB-D observations with language into shared queries, together with learnable temporal embeddings to better retain long-horizon context. Low-level orienting and waypoint navigation are executed to ensure obstacle-aware motion. We further introduce the VLN-Waypoint dataset and a ROS/Gazebo [27] VLN benchmark by migrating Matterport3D [3] scenes into a robotics-native simulator stack. Overall, our contributions are summarized as follows:

- We introduce Hi-Nav, a top-down hierarchical framework that decomposes VLN into high-level task planning, middle-level ISOM-based perception and waypoint reasoning, and low-level execution. By decoupling vision–language reasoning from robot control, Hi-Nav reduces error accumulation and deployment adaptation costs.
- To efficiently realize ISOM-based perception and waypoint reasoning, we propose Hi-VLN, a lightweight multi-task VLN model based on a Q-Former architecture with temporal embeddings to capture historical dependencies, enabling comprehensive fusion of RGB-D and navigation instructions for semantic alignment and robust decision-making in long-horizon navigation.
- Furthermore, We introduce a waypoint reasoning dataset for comprehensive training and establish a ROS/Gazebo VLN benchmark, which significantly

reduces the difficulty of sim-to-real transfer, and extensive experiments in both Gazebo andreal world validate the strong performance of Hi-Nav.

2 Related Work

2.1 VLN in Continuous Environment

Early VLN research [9, 23, 31] is largely established on discretized simulated environments, where the agent navigates by traversing a topological navigation graph (nav-graph) composed of interconnected panoramic viewpoints [2, 4, 20]. Krantz *et al.* [17, 19] reformulated VLN by moving beyond the nav-graph abstraction and instead requiring agents to operate in continuous environments with low-level action execution. In this setting, end-to-end navigation policies directly predict low-level robot actions (e.g., turn left, turn right, move forward, and stop) conditioned on language instructions and visual observations [8, 26, 39, 40, 43], whereas waypoint-based approaches predict an intermediate navigable sub-goal (i.e., the next waypoint) that is subsequently executed by a local controller [7, 11, 24, 33, 37, 41]. With the rapid advances of large-scale vision–language models and the release of increasingly diverse VLN datasets, research on VLN-CE has achieved substantial progress in recent years [34].

2.2 Sim-to-Real Transfer for VLN

VLN models trained in simulation (especially Habitat [25, 29, 30]) often suffer a large performance drop when deployed in the real world (up to 50% reported in [18]). Anderson *et al.* [1] take an important step toward deployment by transferring VLN agents to ROS [27] with subgoal prediction, but their results also highlight how localization drift and execution errors can severely degrade performance. Beyond the nav-graph abstraction, VLN-CE reformulates navigation with low-level actions and later waypoint-based formulations [17, 19]; however, Habitat-style VLN-CE still departs from real robotics in several aspects, including idealized odometry, discrete action primitives that are not directly expressed as continuous velocity commands, and simplified collision handling that cannot capture real failure modes. In parallel, hierarchical navigation has been widely explored to improve long-horizon robustness: a common recipe partitions the environment into topological regions, where a high-level planner selects an ordered sequence of regions to traverse and a low-level controller navigates within each region [10,13,42], while other approaches learn both manager and worker policies via imitation or reinforcement learning [14, 15]. Although effective in controlled settings, these hierarchical methods often rely on environment-specific abstractions or extensive training and do not directly resolve the execution discrepancies that dominate sim-to-real transfer.

In contrast, Hi-Nav is evaluated in Gazebo, a ROS-aligned simulator supporting */cmd_vel*-based control and more realistic execution, and adopts a top-down hierarchical design that separates instruction grounding from motion execution via map-guided perception and waypoint reasoning, yielding improved robustness under real-world constraints.

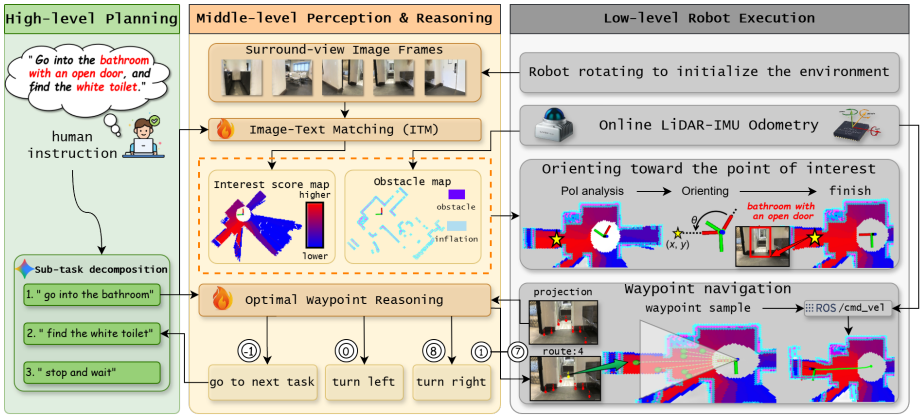

Fig. 1: Pipeline of Hi-Nav. Our hierarchical framework decomposes the complex VLN task into three controllable modules. (1) High-level planning: an LLM-driven planner decomposes the instruction into an ordered sequence of sub-tasks. (2) Middle-level perception and waypoint reasoning: the robot acquires a surround view via in-place rotation, performs image–text matching to build a interest obstacle map, and selects the optimal waypoint to refine the route. (3) Low-level action execution: a ROS-aligned controller executes waypoint navigation through velocity commands (*/cmd_vel*).

3 Method

3.1 Overview

The hierarchical framework of Hi-Nav is illustrated in Fig. 1. We decompose the complex VLN problem in continuous environments into a set of top-down, controllable modules, including high-level task planning, middle-level environment perception and waypoint reasoning, and low-level robot action execution. By explicitly separating robot planning and control from vision–language reasoning, our design is more stable than end-to-end approaches, as it mitigates error accumulation in long-horizon navigation.

3.2 High-level Task Planning

In complex long-horizon VLN tasks, human instructions often encompass multiple primitive commands. Directly processing these long sequences frequently results in ambiguous progress tracking, instruction omission, or premature termination. To mitigate these issues, we propose a High-level Task Planning module that leverages the semantic parsing capabilities of Gemini Pro [32] to decompose complex instructions into fine-grained, executable sub-goals. Our system adopts a closed-loop "decomposition–feedback–iteration" mechanism, in which the mid-level waypoint reasoning module continuously reports sub-task execution status to the high-level planner, and the subsequent sub-task is triggered once the predefined termination conditions are satisfied, thereby improving robustness and adaptability in complex instruction-guided navigation tasks.

3.3 Environment Mapping

Occupancy map. We construct a 3D voxel occupancy map based on log-odds [36] and project it onto a 2D occupancy grid map for subsequent obstacle avoidance and path planning. Specifically, the map takes real-time LiDAR point-cloud as input, which then labels each cell as free, occupied, or unknown via raycasting. The occupancy probability p of each cell is represented by log-odds notation. With the incremental update mechainsm, we maintain a robust online occupancy map, facilitating downstream PoI perception and navigation.

Interest Score Map Generation. To construct the Interest Score Occupancy Map (ISOM) $\mathcal{S} \in [0, 255]^{H \times W}$, we first compute an interest score $s_k \in [0, 1]$ for each surround-view frame I_k. The method for obtaining the score is detailed in Section 3.4. We then project the interest score s_k onto the global occupancy grid to create the ISOM, which is aligned with the obstacle map. The projection is pose-aware: we estimate the robot pose (position and yaw) using LiDAR–IMU odometry [35] and obtain the map-to-robot transform $\mathbf{T}_{r \leftarrow m}$. For each free cell $\mathbf{g}^{\text{free}}$, we compute its cell-center position in the map frame and transform it into the camera frame via calibrated extrinsics, then project it to pixel coordinates using $\mathbf{K}$. Only cells that fall within image bounds and satisfy valid depth are treated as observable. To enforce visibility, we apply a Bresenham ray test from the robot cell to $\mathbf{g}^{\text{free}}$ and suppress scores for occluded cells. For visible cells, we update the ISOM by fusing the previous score with the current evidence using a center-aware weight:

$$\mathcal{S}_{\text{new}}(\mathbf{g}^{\text{free}}) = \frac{\mathcal{S}_{\text{old}}(\mathbf{g}^{\text{free}}) + \lambda\, s_k\, w(u)}{2}, \quad w(u) = \Big(1 - \frac{|u - c_x|}{W_I/2}\Big)^{\alpha}, \tag{1}$$

where $\lambda = 255$, and $w(u)$ downweights evidence near image boundaries.

PoI Analysis. To robustly localize a point of interest (PoI) from the accumulated ISOM, we perform a lightweight hotspot analysis on the cached average map $\mathcal{S}$. We first apply a small Gaussian smoothing to suppress spurious peaks, then compute a percentile-based threshold over non-zero cells to select high-confidence hotspot candidates (i.e., the top-p fraction of scores). Given the candidate set, we estimate the PoI as a weighted centroid, where each cell is weighted by its score raised to a power γ to emphasize consistently salient regions while remaining less sensitive to single-frame outliers. Finally, the PoI is converted from grid coordinates to world coordinates using the map resolution and origin pose, and published for downstream robot orienting.

3.4 Environmental Perception and Waypoint Navigation

Model Architecture As illustrated in Fig. 2, we propose Hi-VLN, a lightweight multi-task framework that jointly addresses Environment Perception and Waypoint Navigation within a unified architecture. The system incrementally con-

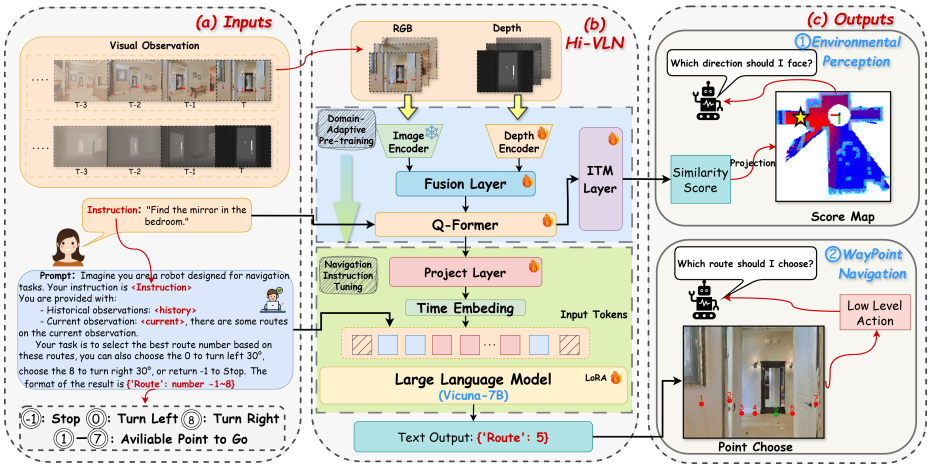

Fig. 2: Overview of the proposed Hi-VLN architecture. The system comprises three stages: (a) Inputs, which integrates historical and current RGB-D observations with prompt-formatted instructions; (b) Hi-VLN, the core reasoning engine that aligns dual-stream visual features via a Q-Former and makes decisions using a finetuned LLM (Vicuna-7B); and (c) Outputs, which simultaneously performs Environmental Perception via semantic score maps and Waypoint Navigation by translating text decisions into low-level physical actions.

structs a multimodal contextual representation from historical RGB-D observations and structured language prompts.To enhance spatial awareness, a dual-stream encoding mechanism first extracts visual appearance features from RGB images and structural depth cues from the depth map. A dedicated cross-attention network is then employed to integrate RGB and depth features, enabling fine-grained cross-modal interaction and strengthening spatial representations.The fused visual features are subsequently aligned with the navigation instruction through a domain-adaptive Q-Former [22], which encodes visual and textual modalities into compact, instruction-aware query embeddings. These query embeddings are routed to a decoupled dual-branch decision module.

In the perception branch, an Image-Text Matching (ITM) head operates on the unified queries to estimate the semantic alignment score between the instruction and the current observation. The predicted matching scores are further projected to construct the ISOM, enabling explicit perception of instruction-relevant regions within the environment. In parallel, the action branch enriches the query embeddings with temporal sequence information, projects the resulting representations into a sequence of tokens, and feeds them into a LoRA-finetuned large language model (Vicuna-7B [6]), which autoregressively predicts optimal waypoints and low-level navigation actions. Through this tightly coupled yet functionally decoupled design, Hi-VLN achieves explicit environment perception and robust decision-making within a single unified framework, while maintaining computational efficiency suitable for edge-device deployment.

Training Strategy We adopt a two-stage training paradigm to ensure stable convergence and effective cross-modal learning. In the domain-adaptive pretraining stage, the RGB encoder is frozen, while the Depth Encoder, Fusion Layer, and ITM-Head Layer are trained to align multi-modal geometric observations with language semantics, establishing explicit environmental perception. This stage is supervised using a binary cross-entropy loss on the publicly available Flickr30K [38] dataset.

In the navigation instruction tuning stage, the perception modules are fixed, and the Projection Layer together with the Vicuna-7B LLM is optimized via LoRA. Using structured prompts that incorporate historical observations and candidate waypoints, the action branch learns to perform instruction-aware spatial reasoning and autoregressively predict optimal waypoint navigation actions. To train this stage, we employ a novel loss function on our self-constructed dataset(VLN-Waypoint). This loss integrates a cross-entropy-based hard-label supervision term with a spatial distance-aware soft-label loss.

Specifically, we formulate waypoint navigation as an autoregressive token prediction task optimized via a cross-entropy loss. For a sequence of length T, the hard loss $\mathcal{L}_{hard}$ is defined as:

$$\mathcal{L}_{hard} = -\frac{1}{T}\sum_{t=1}^{T}\log P(x_t \mid x_{<t}, \mathbf{V}_{hist}, \mathbf{V}_{curr}), \tag{2}$$

where x_t denotes the target token at step t, and $\mathbf{V}_{hist}$ and $\mathbf{V}_{curr}$ represent the historical and current visual features, respectively.

To further enhance spatial awareness and penalize predictions according to their topological distance from the optimal waypoint, we introduce a distance-aware soft loss $\mathcal{L}_{soft}$. This term minimizes the Kullback–Leibler (KL) divergence between the predicted navigation probability distribution $Pnav$ and a Gaussian target distribution Q:

$$\mathcal{L}_{soft} = \sum_{k=0}^{n} Q(k)\log\left(\frac{Q(k)}{P_{nav}(k)}\right). \tag{3}$$

where n denotes the total number of candidate navigation waypoints.

Finally, the overall objective function for the navigation module is formulated as a weighted sum of these components:

$$\mathcal{L}_{nav} = \mathcal{L}_{hard} + \lambda \cdot \mathcal{L}_{soft}, \tag{4}$$

where λ is a hyperparameter designed to balance the contributions of the exact token prediction and the soft spatial regularization.

VLN–Waypoint Dataset Construction To bridge the gap between navgraph VLN supervision and waypoint-driven continuous navigation, we construct a VLN–Waypoint dataset that augments VLN trajectories with waypoint candidates and discrete waypoint labels (Fig. 3). Starting from VLN–ego data [26]

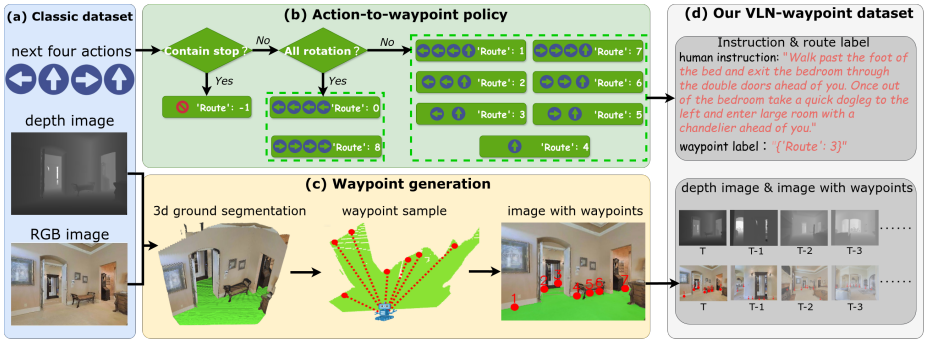

Fig. 3: VLN–Waypoint dataset construction pipeline. (a) Starting from classic VLN observations (RGB/depth) and multi-step action sequences. (b) An action-to-waypoint policy maps the next four actions to a discrete waypoint label (including *stop*). (c) Using depth-based 3D ground segmentation, we sample traversable waypoint candidates and project them into the image to obtain waypoint-annotated views. (d) We package each sample as an instruction–route pair with current and history frames, together with the projected waypoint candidates, forming our VLN–Waypoint dataset.

with RGB observations and low-level expert actions, we additionally render synchronized depth maps in Habitat. Following VLMNav [11] and WMNav [24], we back-project RGB-D into colored point clouds, segment ground points (using patchwork [21]), and sample traversable waypoint candidates around the agent. These 3D candidates are projected onto the image to obtain waypoint-annotated views. We then map the next four robot actions to a single waypoint label via an action-to-waypoint policy. Specifically, we discretize the image into ordered waypoint slots and use a short-horizon action sequence to determine the route label: if the first action is *stop*, we assign a terminal label (-1). Otherwise, we scan the sequence until the first *forward* action and count the number of left/right turns. Pure turning sequences are mapped to extreme waypoints, i.e., *all-left* $\rightarrow$ label 0 and *all-right* $\rightarrow$ label 8. An *all-forward* sequence maps to the center waypoint (label 4). For mixed sequences, the label is selected by the turn count before moving forward: one/two/three left turns map to 3/2/1, and one/two/three right turns map to 5/6/7. We finally package each sample as an instruction–route pair with current/history frames and the projected candidates, yielding the VLN–Waypoint dataset.

3.5 Robot Action Execution

Orienting toward the point of interest After localizing the point of interest (PoI) in the world frame, $\mathbf{p}_{poi} = [x_{poi}, y_{poi}]^\top$, we orient the robot to face it. Given the robot pose $\mathbf{p}_r = [x_r, y_r]^\top$ and yaw θ_r, the desired heading is

$$\theta_d = \text{atan2}(y_{poi} - y_r,\ x_{poi} - x_r). \tag{5}$$

A local controller performs an in-place rotation to minimize the angular error between θ_r and θ_d until it falls within a tolerance.

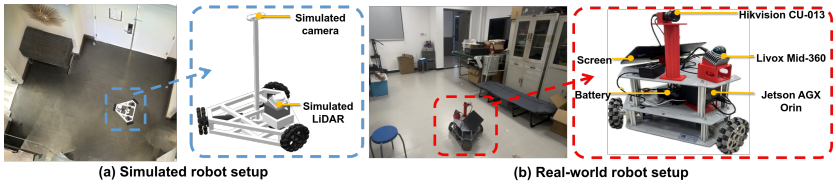

Fig. 4: Robot setup in (a) simulation environment and (b) real-world environment.

Waypoint navigation We execute waypoint commands using a standard ROS navigation stack. The global planner runs the A* algorithm [12] on the costmap to compute a feasible global path, and the local planner (TEB) [28] tracks the path while satisfying kinematic constraints and avoiding obstacles.

4 Experiments

4.1 Experimental Settings

Simulation Environment. We evaluate the proposed Hi-VLN model on the R2R dataset [2] in VLN-CE [19], which contains 90 Matterport3D [3] indoor scenes and 16,844 path–instruction pairs. We further migrate Matterport3D scenes into Gazebo [16] and design Easy, Medium, and Hard navigation tasks to evaluate navigation performance. The simulation environment is shown in Fig. 4(a). Since VLN-CE assumes idealized odometry and discrete actions, only the Hi-VLN module is evaluated under the standard protocol. The complete Hi-Nav system is therefore primarily validated in Gazebo and real-world.

Real-world Environment. We evaluate navigation tasks of varying difficulty including single-task and multi-subtask settings. We use a Wheeltech omnidirectional mobile robot platform as the experimental carrier, equipped with an industrial camera, LiDAR, and edge computing devices (see Fig. 4(b) for details).

Metrics. We adopt standard metrics for evaluating navigation performance [2, 19, 20], including Success Rate (SR), Navigation Error (NE), Success weighted by Path Length (SPL), and Oracle Success (OS).

Baselines. We compare against representative continuous VLN baselines, including CMA [19], Seq2Seq [19], WS-MGMap [5], NaVid [40], and Uni-NaVid [39]. To ensure a fair comparison, we integrate these methods into the ROS system.

4.2 Model Performance

Since our VLN model is designed as a multi-task architecture, we conduct separate performance tests for each proposed module. We first validate the ITM module's efficacy in visual-linguistic alignment, as shown in Fig. 5. Furthermore,

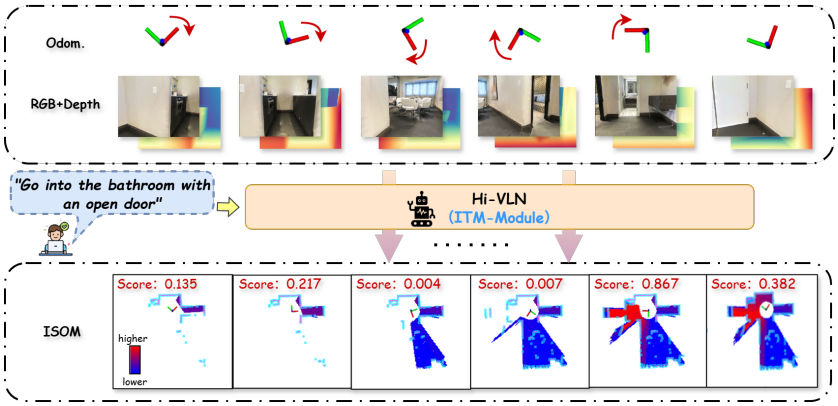

Fig. 5: Visualization result of the ITM-module. The agent captures RGB-D images at each viewpoint. Conditioned on the given instruction, it sequentially computes matching scores and projects them onto ISOM, ultimately constructing a complete map.

Table 1: Comparison on VLN-CE R2R Val-Unseen.

	Observation			VLN-CE R2R Val-Unseen				
Method	RGB	Depth	Odom.	TL	NE↓	OS↑	SR↑	SPL↑
CMA [19]	✓	✓		8.64	7.37	40.0	32.0	30.0
Seq2Seq [19]	✓	✓		9.30	7.77	37.0	25.0	22.0
WS-MGMap [5]	✓	✓	✓	10.00	6.28	47.6	38.9	34.3
NaviD [40]	✓			7.63	5.47	49.1	37.4	35.9
Uni-NaviD [39]	✓			9.71	5.58	**53.3**	47.0	42.7
Hi-VLN(ours)	✓	✓		9.62	**4.47**	51.3	**47.2**	**43.1**

to specifically evaluate the model's capability in waypoint-based navigation, we conduct comparative experiments on the VLN-CE benchmark without employing the ITM module. For a fair comparison, both the baseline methods and our proposed model are trained on the official training split, which includes 10,819 R2R [2] samples, and subsequently evaluated on the validation split consisting of 1,839 R2R samples. The quantitative results are presented in Table 1.

Additionally, we conduct an ablation study to evaluate the impact of various components within our proposed architecture on waypoint-based navigation accuracy(Table 2). Experimental results on the VLN-CE R2R validation unseen set show that the full model achieves superior performance.

4.3 Gazebo Navigation Experiment

We imported 90 indoor scenes from the Matterport3D dataset into the Gazebo simulation platform and designed navigation tasks with three difficulty levels (Easy, Medium, and Hard) to evaluate the navigation performance of our proposed Hi-Nav VLN framework in Gazebo. For each method, we run 20 trials at each difficulty level to obtain statistically meaningful results. Specifically,

Table 2: Ablation study of the proposed components on the VLN-CE R2R validation unseen set. We evaluate the individual and synergistic contributions of the Depth encoder (Depth), Cross-Attention mechanism (CA), and Time Embedding (Time). Model 1 serves as the baseline, while Model 4 represents our full architecture.

Model	Components			Metrics				
	Depth	CA	Time	TL	NE↓	OS↑	SR↑	SPL↑
1				10.45	7.75	24.5	21.8	18.5
2	✓			10.64	7.64	28.9	22.8	18.9
3	✓	✓		9.60	5.32	45.8	40.9	34.3
4	✓	✓	✓	9.84	**4.32**	**51.6**	**47.3**	**42.5**

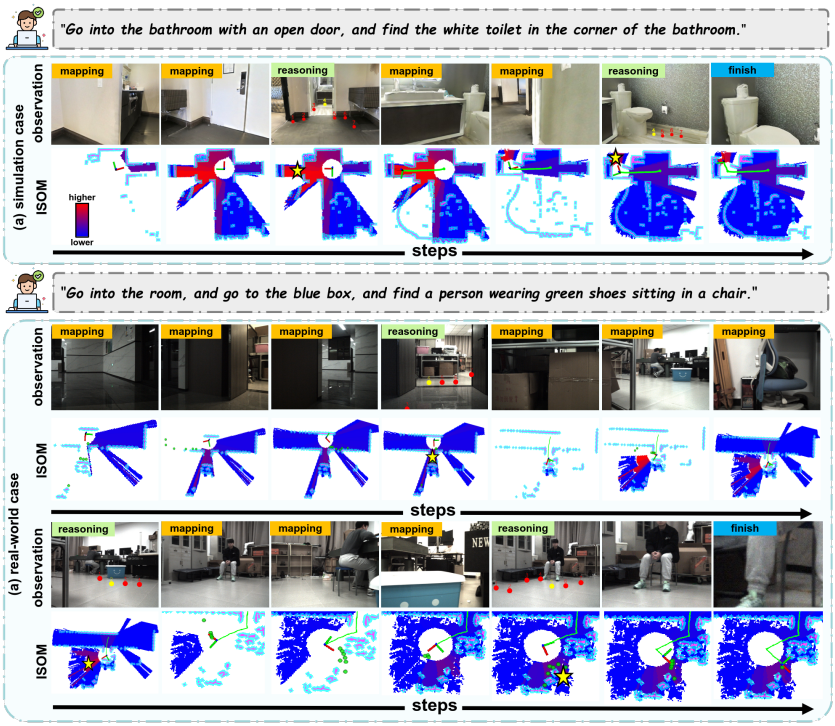

Fig. 6: We visualize intermediate results of our method while executing navigation tasks in both simulation environment and real-world settings. The observation row shows the RGB images captured by the robot, and the images annotated with Reasoning indicate the waypoint reasoning results. The ISOM row shows the interest score occupancy map built during navigation, where the green line denotes the trajectory.

easy tasks place the goal within the agent's initial field of view, middle tasks require searching as the goal is initially out of view, and hard tasks involve long-horizon, multi-stage instructions with sequential sub-goals. As shown by the intermediate results in Fig. 6, leveraging the hierarchical design, Hi-Nav first infers the bathroom direction through environment perception and then uses waypoint reasoning to select an optimal route to enter the bathroom. It re-

Table 3: Performance comparison of different methods across difficulty levels in the Gazebo simulation environment.

		Observation			Easy		Middle		Hard		Overall	
Method	End2End	RGB	Depth	LiDAR	SR↑	NE↓	SR↑	NE↓	SR↑	NE↓	SR↑	NE↓
CMA [19]	✓	✓	✓		0.0	5.56	0.0	6.14	0.0	6.69	0.0	6.13
Seq2Seq [19]	✓	✓	✓		5.0	5.81	0.0	6.45	0.0	6.97	1.7	6.41
NaviD [40]	✓	✓			55.0	2.60	25.0	3.88	10.0	5.36	30.0	3.95
Uni-NaviD [39]	✓	✓			45.0	2.37	35.0	3.67	20.0	5.11	33.3	3.72
VLFM [37]		✓	✓	✓	75.0	4.79	15.0	5.69	0.0	10.43	30.0	6.97
Hi-Nav(ours)		✓	✓	✓	**90.0**	**0.48**	**80.0**	**0.42**	**70.0**	**1.05**	**80.0**	**0.65**

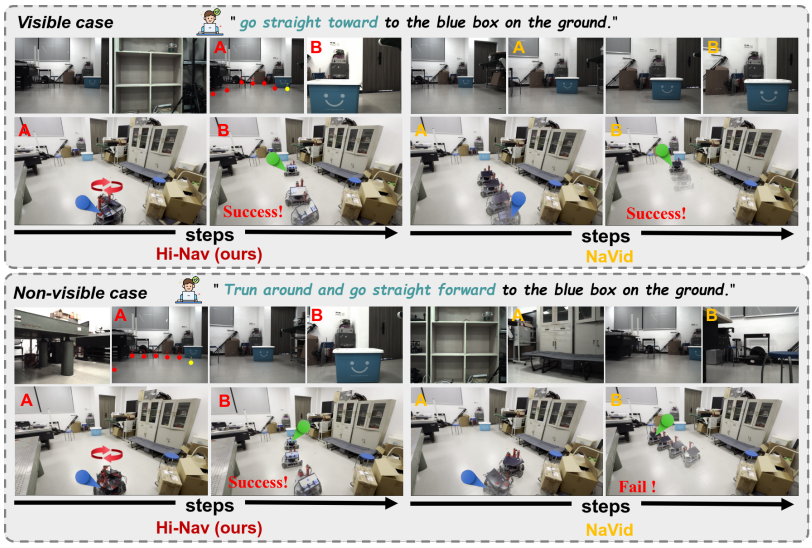

Fig. 7: Qualitative comparison between Hi-Nav and NaVid [40] on real-world navigation. Top: Visible case. The target (blue box) is within the initial view. Bottom: Non-visible case. The target is initially out of view. Markers A and B denote representative viewpoints. The colored pins indicate the start and goal locations.

peats this perception–reasoning–execution loop across sub-tasks and ultimately reaches the target beside the toilet.

Table 3 reports the performance comparison across the three difficulty levels. The results show that our method consistently outperforms all baseline approaches by a clear margin on every difficulty level, validating the effectiveness of our hierarchical framework in a more realistic simulation environment.

4.4 Real-world Navigation Experiment

Visible vs. Non-visible Goal. In this section, we conducted experiments to evaluate the navigation performance under two settings: when the goal is visible from the start and when it is not. As shown in Fig. 7, in the visible case (top), both Hi-Nav and NaVid successfully reach the blue box, suggesting that direct visual grounding is often sufficient when the target is within the initial field of

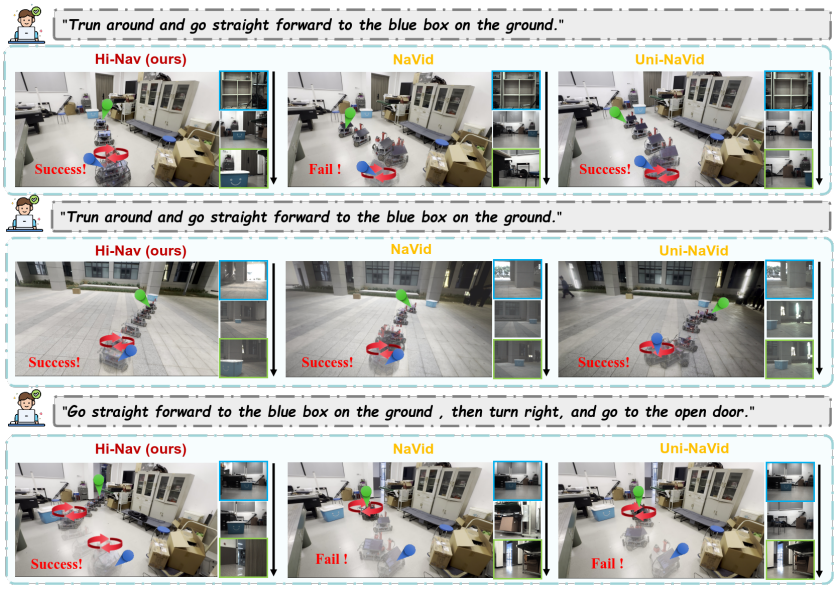

Fig. 8: Real-world qualitative comparison of Hi-Nav (ours), NaVid [40], and Uni-NaVid [39] on single-goal and multi-subtask instructions. The colored pins indicate the start and goal locations.

Table 4: Real-world performance under different goal visibility and goal-count settings.

	Visible						Non-visible					
	1-goal		2-goal		3-goal		1-goal		2-goal		3-goal	
Method	SR↑	NE↓	SR↑	NE↓	SR↑	NE↓	SR↑	NE↓	SR↑	NE↓	SR↑	NE↓
NaviD [40]	60.0	1.51	35.0	4.72	0.0	6.24	25.0	3.97	0.0	5.13	0.0	4.71
Uni-NaviD [39]	70.0	2.02	15.0	3.02	0.0	5.80	45.0	2.85	0.0	5.82	0.0	8.44
Hi-Nav(ours)	**95.0**	**0.49**	**55.0**	**1.29**	**25.0**	**1.14**	**85.0**	**0.42**	**55.0**	**0.97**	**35.0**	**1.06**

view. In contrast, the non-visible case (bottom) is substantially more challenging due to partial observability: the robot must first reorient to discover informative views before making forward progress. Benefiting from our hierarchical design, Hi-Nav explicitly performs an in-place rotation to acquire a surround view (red circular arrows), leverages the ISOM to localize the point of interest, and then refines the route via waypoint reasoning, leading to a successful arrival at the target. NaVid fails in this setting, as the end-to-end policy tends to commit to locally plausible motions without reliable re-orientation and lacks an explicit waypoint refinement to recover once it moves away from informative viewpoints.

Single-task vs. Multi-subtask. We further study instruction compositionality by comparing single-task and multi-subtask navigation on a real robot. As shown in Fig. 8, the first two rows correspond to single-goal commands ("*turn around and go straight to the blue box*") in indoor and outdoor scenes. These cases are largely solvable once the target is grounded; all methods succeed outdoors,

Table 5: Ablation study in Gazebo across difficulty levels (20 trials each). We evaluate the impact of sub-task decomposition (SD) and ISOM-based map guidance on success rate (SR) and navigation error (NE).

			Easy		Medium		Hard		Overall	
Setting	SD	ISOM	SR↑	NE↓	SR↑	NE↓	SR↑	NE↓	SR↑	NE↓
1			70.0	1.20	5.0	13.79	0.0	16.28	25.0	10.42
2		✓	80.0	1.71	80.0	2.40	0.0	14.09	53.3	6.07
3	✓		70.0	2.63	5.0	8.29	5.0	2.60	26.7	4.50
4 (ours)	✓	✓	**90.0**	**0.48**	**80.0**	**0.42**	**70.0**	**1.05**	**80.0**	**0.65**

while NaVid fails in the cluttered indoor scene, indicating higher sensitivity to partial observability and local ambiguities. Even when NaVid and Uni-NaVid succeed, they often take less direct routes and incur larger NE than Hi-Nav. The third row presents a multi-subtask instruction that requires sequentially reaching the blue box and then the open door. This setting requires progress tracking and re-grounding after the first goal. Hi-Nav completes both objectives by executing sub-tasks with map-guided re-orientation and waypoint refinement, whereas NaVid and Uni-NaVid fail to reach the second target.

Table 4 confirms these trends over 20 runs per setting. Hi-Nav achieves consistently higher success rates with smaller NE across different goal visibility and goal counts, and its advantage becomes more pronounced as the task grows harder, particularly for non-visible and multi-goal instructions.

4.5 Ablation Study for Hi-Nav

Table 5 shows that SD and ISOM play complementary roles. Without either component (Setting 1), the agent performs reasonably on Easy but largely fails on Medium/Hard with very large NE, highlighting brittleness under partial observability and long horizons. Adding ISOM alone (Setting 2) markedly boosts Easy and Medium success, indicating that semantic–geometric map guidance is effective for orienting and local decisions, yet it still fails on Hard. Using SD without ISOM (Setting 3) provides limited gains, suggesting that sub-task structuring cannot replace reliable map-guided localization. When SD and ISOM are combined (Setting 4), performance improves substantially across all levels with low NE, demonstrating their strong synergy, especially for long-horizon tasks.

5 Conclusion and Limitation

In this work, we propose Hi-Nav, a hierarchical framework for vision-and-language navigation in continuous environments, and demonstrate its effectiveness in both Gazebo simulation and real-world robot experiments. Our approach improves robustness for long-horizon and partially observable navigation via structured, map-guided decision making. Limitations include sensitivity to ambiguous instructions and sensing noise, as well as limited diversity of environments and instruction types in our current benchmark.

6 Acknowledgement

The numerical calculations in this article had been supported by the supercomputing system in the Supercomputing Center of Wuhan University, Wuhan, China. This research is supported by the Fundamental Research Funds for the Central Universities-Key Interdisciplinary Team Development Project (No. 2042026kf0018).

References

1. Anderson, P., Shrivastava, A., Truong, J., Majumdar, A., Parikh, D., Batra, D., Lee, S.: Sim-to-Real Transfer for Vision-and-Language Navigation. In: Proceedings of the 2020 Conference on Robot Learning. pp. 671–681. PMLR
2. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., van den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
3. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from RGB-D data in indoor environments. International Conference on 3D Vision (3DV) (2017)
4. Chen, H., Suhr, A., Misra, D., Snavely, N., Artzi, Y.: Touchdown: Natural language navigation and spatial reasoning in visual street environments. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12530–12539 (2019). https://doi.org/10.1109/CVPR.2019.01282
5. Chen, P., Ji, D., Lin, K., Zeng, R., Li, T., Tan, M., Gan, C.: Weakly-supervised multi-granularity map learning for vision-and-language navigation. Advances in Neural Information Processing Systems **35**, 38149–38161 (2022)
6. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., et al.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See https://vicuna. lmsys. org (accessed 14 April 2023) **2**(3), 6 (2023)
7. Du, Y., Fu, T., Chen, Z., Li, B., Su, S., Zhao, Z., Wang, C.: Vl-nav: Real-time vision-language navigation with spatial reasoning (2025), https://arxiv.org/abs/2502.00931
8. Fan, S., Liu, R., Wang, W., Yang, Y.: Navigation instruction generation with bev perception and large language models. In: ECCV (2024)
9. Fried, D., Hu, R., Cirik, V., Rohrbach, A., Andreas, J., Morency, L.P., Berg-Kirkpatrick, T., Saenko, K., Klein, D., Darrell, T.: Speaker-follower models for vision-and-language navigation. In: Neural Information Processing Systems (NeurIPS) (2018)
10. Gao, C., Peng, X., Yan, M., Wang, H., Yang, L., Ren, H., Li, H., Liu, S.: Adaptive zone-aware hierarchical planner for vision-language navigation. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14911–14920 (2023). https://doi.org/10.1109/CVPR52729.2023.01432
11. Goetting, D., Singh, H.G., Loquercio, A.: End-to-end navigation with vlms: Transforming spatial reasoning into question-answering. In: Workshop on Language and Robot Learning: Language as an Interface (2024)

12. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics **4**(2), 100–107 (1968). https://doi.org/10.1109/TSSC.1968.300136
13. He, Z., Wang, N., Wang, L., Liu, C., Chen, Q.: Instruction-aligned hierarchical waypoint planner for vision-and-language navigation in continuous environments. Pattern Analysis and Applications **27**(4), 132 (Oct 2024). https://doi.org/10.1007/s10044-024-01339-z, https://doi.org/10.1007/s10044-024-01339-z
14. Irshad, M.Z., Ma, C.Y., Kira, Z.: Hierarchical cross-modal agent for robotics vision-and-language navigation. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 13238–13246 (2021). https://doi.org/10.1109/ICRA48506.2021.9561806
15. Johnson, F., Cao, B.B., Ashok, A., Jain, S., Dana, K.: Feudal networks for visual navigation (2024), https://arxiv.org/abs/2402.12498
16. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566). vol. 3, pp. 2149–2154 vol.3 (2004). https://doi.org/10.1109/IROS.2004.1389727
17. Krantz, J., Gokaslan, A., Batra, D., Lee, S., Maksymets, O.: Waypoint Models for Instruction-guided Navigation in Continuous Environments. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 15142–15151. IEEE
18. Krantz, J., Lee, S.: Sim-2-sim transfer for vision-and-language navigation in continuous environments. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022. pp. 588–603. Springer Nature Switzerland
19. Krantz, J., Wijmans, E., Majundar, A., Batra, D., Lee, S.: Beyond the nav-graph: Vision and language navigation in continuous environments. In: European Conference on Computer Vision (ECCV) (2020)
20. Ku, A., Anderson, P., Patel, R., Ie, E., Baldridge, J.: Room-Across-Room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In: Conference on Empirical Methods for Natural Language Processing (EMNLP) (2020)
21. Lee, S., Lim, H., Myung, H.: Patchwork++: Fast and robust ground segmentation solving partial under-segmentation using 3D point cloud. In: Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. pp. 13276–13283 (2022)
22. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: International conference on machine learning. pp. 19730–19742. PMLR (2023)
23. Ma, C.Y., Lu, J., Wu, Z., AlRegib, G., Kira, Z., Socher, R., Xiong, C.: Self-monitoring navigation agent via auxiliary progress estimation. In: Proceedings of the International Conference on Learning Representations (ICLR) (2019), https://arxiv.org/abs/1901.03035
24. Nie, D., Guo, X., Duan, Y., Zhang, R., Chen, L.: Wmnav: Integrating vision-language models into world models for object goal navigation. arXiv preprint arXiv:2503.02247 (2025)
25. Puig, X., Undersander, E., Szot, A., Cote, M.D., Partsey, R., Yang, J., Desai, R., Clegg, A.W., Hlavac, M., Min, T., Gervet, T., Vondrus, V., Berges, V.P., Turner, J., Maksymets, O., Kira, Z., Kalakrishnan, M., Malik, J., Chaplot, D.S., Jain, U., Batra, D., Rai, A., Mottaghi, R.: Habitat 3.0: A co-habitat for humans, avatars and robots (2023)
26. Qi, Z., Zhang, Z., Yu, Y., Wang, J., Zhao, H.: Vln-r1: Vision-language navigation via reinforcement fine-tuning. arXiv: 2506.17221 (2025)

27. Quigley, M., Gerkey, B.P., Conley, K., Faust, J., Ng, A.Y.: Ros: An open-source robot operating system (2009)
28. Rösmann, C., Hoffmann, F., Bertram, T.: Integrated online trajectory planning and optimization in distinctive topologies. Robotics and Autonomous Systems **88**, 142–153 (2017)
29. Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., Parikh, D., Batra, D.: Habitat: A platform for embodied ai research. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9338–9346 (2019). https://doi.org/10.1109/ICCV.2019.00943
30. Szot, A., Clegg, A., Undersander, E., Wijmans, E., Zhao, Y., Turner, J., Maestre, N., Mukadam, M., Chaplot, D., Maksymets, O., Gokaslan, A., Vondrus, V., Dharur, S., Meier, F., Galuba, W., Chang, A., Kira, Z., Koltun, V., Malik, J., Savva, M., Batra, D.: Habitat 2.0: Training home assistants to rearrange their habitat. In: Advances in Neural Information Processing Systems (NeurIPS) (2021)
31. Tan, H., Yu, L., Bansal, M.: Learning to navigate unseen environments: Back translation with environmental dropout. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 2610–2621. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1268, https://aclanthology.org/N19-1268/
32. Team, G.: Gemini: A family of highly capable multimodal models (2025), https://arxiv.org/abs/2312.11805
33. Wang, L., He, Z., Li, J., Xia, R., Hu, M., Yao, C., Liu, C., Tang, Y., Chen, Q.: Clash: Collaborative large-small hierarchical framework for continuous vision-and-language navigation (2026), https://arxiv.org/abs/2512.10360
34. Wang, Z., Li, J., Hong, Y., Wang, Y., Wu, Q., Bansal, M., Gould, S., Tan, H., Qiao, Y.: Scaling data generation in vision-and-language navigation. In: ICCV 2023 (2023)
35. Xu, W., Cai, Y., He, D., Lin, J., Zhang, F.: Fast-lio2: Fast direct lidar-inertial odometry. IEEE Transactions on Robotics **38**(4), 2053–2073 (2022). https://doi.org/10.1109/TRO.2022.3141876
36. Xu, Z., Zhan, X., Chen, B., Xiu, Y., Yang, C., Shimada, K.: A real-time dynamic obstacle tracking and mapping system for uav navigation and collision avoidance with an rgb-d camera. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 10645–10651 (2023). https://doi.org/10.1109/ICRA48891.2023.10161194
37. Yokoyama, N., Ha, S., Batra, D., Wang, J., Bucher, B.: Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In: International Conference on Robotics and Automation (ICRA) (2024)
38. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the association for computational linguistics **2**, 67–78 (2014)
39. Zhang, J., Wang, K., Wang, S., Li, M., Liu, H., Wei, S., Wang, Z., Zhang, Z., Wang, H.: Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks (2024)
40. Zhang, J., Wang, K., Xu, R., Zhou, G., Hong, Y., Fang, X., Wu, Q., Zhang, Z., Wang, H.: Navid: Video-based vlm plans the next step for vision-and-language navigation. Robotics: Science and Systems (2024)

41. Zhang, M., Du, Y., Wu, C., Zhou, J., Qi, Z., Ma, J., Zhou, B.: Apexnav: An adaptive exploration strategy for zero-shot object navigation with target-centric semantic fusion. IEEE Robotics and Automation Letters **10**(11), 11530–11537 (2025). https://doi.org/10.1109/LRA.2025.3606388
42. Zhang, S., Song, X., Yu, X., Bai, Y., Guo, X., Li, W., Jiang, S.: Hoz++: Versatile hierarchical object-to-zone graph for object navigation. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**(7), 5958–5975 (2025). https://doi.org/10.1109/TPAMI.2025.3552987
43. Zhou, G., Hong, Y., Wang, Z., Wang, X.E., Wu, Q.: Navgpt-2: Unleashing navigational reasoning capability for large vision-language models. In: Computer Vision – ECCV 2024. pp. 260–278. Springer Nature Switzerland, Cham (2025)