Three-Step Nav: A Hierarchical Global–Local Planner for Zero-Shot Vision-and-Language Navigation

Anonymous Author
Anonymous Institution

Abstract

Breakthrough progress in vision-based navigation through unknown environments has been achieved by using multimodal large language models (MLLMs). These models can plan a sequence of motions by evaluating the current view at each time step against the task and goal given to the agent. However, current zero-shot Vision-and-Language Navigation (VLN) agents powered by MLLMs still tend to drift off course, halt prematurely, and achieve low overall success rates. We propose Three-Step Nav to counteract these failures with a three-view protocol: First, "look forward" to extract global landmarks and sketch a coarse plan. Then, "look now" to align the current visual observation with the next sub-goal for fine-grained guidance. Finally, "look backward" audits the entire trajectory to correct accumulated drift before stopping. Requiring no gradient updates or task-specific fine-tuning, our planner drops into existing VLN pipelines with minimal overhead. Three-Step Nav achieves state-of-the-art zero-shot performance on the R2R-CE and RxR-CE dataset.

1 Introduction

In the Vision-and-Language Navigation (VLN) (Anderson et al., 2018; Gu et al., 2022; Wang et al., 2023a; Li et al., 2024) task, embodied agents are required to navigate to an unseen destination following a series of natural language instructions. Early studies (An et al., 2024; Zhou et al., 2024a) simplified the task by grounding navigation in discrete graphs, where the agent selects between predefined viewpoints and edges depending on RGB-D sensory inputs. Although effective for benchmarking, this abstraction neglects the low-level dynamics of real-world robotics, such as continuous motion, partial observability, and collision risks. To bridge this gap, the community has introduced VLN in Continuous Environments (VLN-CE) (Krantz et al., 2020), which eliminates the dependence on connectivity graphs and instead equips agents with egocentric

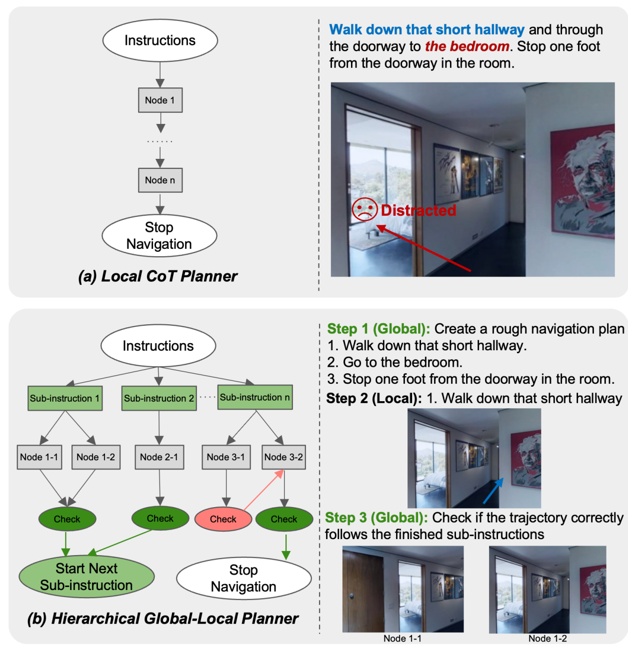

Figure 1: (a) Prior LLM-core planners rely only on the current RGB-D view and textual action history, often misjudging progress and getting distracted by irrelevant objects. In contrast, (b) our Hierarchical Global–Local Planner mitigates this by first decomposing instructions into sub-instructions for a global plan, then locally grounding each sub-instruction, and finally verifying completed sub-instructions against the global trajectory—yielding more robust zero-shot navigation.

sensors and low-level actions. This setting is substantially more challenging than discrete graph-based VLN, as it requires reasoning over open spaces, handling unforeseen obstacles, and maintaining progress without predefined navigation graphs. Building agents that can solve VLN-CE reliably represents a critical step toward general-purpose embodied AI, enabling real-world applications such as home robotics, assistive navigation, and search and rescue operations.

Meanwhile, Multimodal Large Language Models (MLLMs) have demonstrated strong zero-shot abilities in vision-language tasks (Zhou et al., 2024b; Long et al., 2024; Chen et al., 2024). Their broad world knowledge and flexible reasoning suggest a new path for VLN: treat navigation as an iterative decision-making dialogue with a powerful but generic model, rather than training a bespoke policy from scratch. However, directly requiring an MLLM faces two key challenges in continuous VLN: As shown in Fig. 1 (a), instructions can span dozens of steps; the agent must balance global route planning with local actuation, yet commodity MLLMs reason over a limited context window. Moreover, in continuous space, small heading or position errors quickly compound. The agent needs principled mechanisms to detect mistakes and recover without explicit supervision.

To address these issues, we present *Three-Step Nav*, a hierarchical global–local framework that leverages MLLMs for zero-shot VLN in continuous 3-D environments, differing from prior MLLM-based VLN agents by coupling global planning with trajectory-level verification. The agent alternates a global–local–global reasoning loop: (i) *looking forward* to outline upcoming sub-instructions, (ii) *looking now* to ground the current sub-goal in live visual observations, and (iii) *looking back* to verify past progress and adjust future plans. Within this loop, we introduce an *adaptive judge module* that endows the agent with four meta-skills - *stay*, *continue*, *backtrack*, and *look-around* - allowing dynamic self-correction when uncertainty is detected.

Compared with prior LLM-core planners, our hierarchical global–local design achieves state-of-the-art zero-shot success rates R2R-CE (Ku et al., 2020) and RxR-CE (Krantz et al., 2020) datasets, while also reducing navigation error by 15% and improving SPL by 12% on the validation-unseen splits of R2R-CE, indicating that our global progress check and trajectory-level auditing effectively mitigate distraction and cumulative drift in continuous environments. The goals of this work can be summarized as follows:

- We propose Three-Step Nav, a novel framework that alternates global–local–global reasoning with an MLLM, enabling zero-shot VLN in continuous 3-D environments while preserving long-range context and requiring no task-specific fine-tuning. Moreover, the framework is lightweight and modular, making it easy to plug into other LLM-core planning pipelines.
- We introduce a Hierarchical Global–Local Planner that dynamically switches views: after completing local chain-of-thought reasoning to reach a specific sub-goal, the agent transitions to a global check that examines the trajectory and verifies finished sub-instructions. This alternating loop between fine-grained execution and trajectory-level auditing helps the MLLM reason over spatial structure and exploration history, suppressing distraction and cumulative drift.
- We design an adaptive judge module equipped with meta-skills to decide whether to *stay*, *continue*, *backtrack*, or *look around*, allowing the agent to self-correct under uncertainty and maintain robust navigation even in ambiguous environments.

2 Related Work

Vision-and-Language Navigation The VLN task (Anderson et al., 2018) requires embodied agents to follow natural language instructions and visual observations to reach goal locations in novel environments. Early research in the discrete VLN setting, typically built on the Matterport3D simulator (Chang et al., 2017), modeled navigation as sequential decision making on a predefined connectivity graph. To enhance performance, methods introduced various structural priors: DUET (Chen et al., 2022) and ETPNav (An et al., 2024) leveraged topological abstractions to capture global spatial relations, BEVBert (An et al., 2023) constructed semantic bird's-eye view representations, and Wang et al. (2023b) designed egocentric grid-based memory for long-horizon reasoning. Beyond graph-based or memory-driven methods, VLN-Video (Li et al., 2024) utilized driving videos to extend VLN into outdoor navigation, and Zhu et al. (2023) introduced knowledge-driven imagination of unseen layouts. Collectively, these approaches advanced discrete VLN by enriching agents' spatial reasoning and grounding, laying the foundation for more realistic navigation formulations. Different from these training-intensive approaches, our work explores training-free, zero-shot VLN framework, which employing MLLMs as core planners.

Navigation with MLLM LLMs have demonstrated remarkable generalization and reasoning capa-

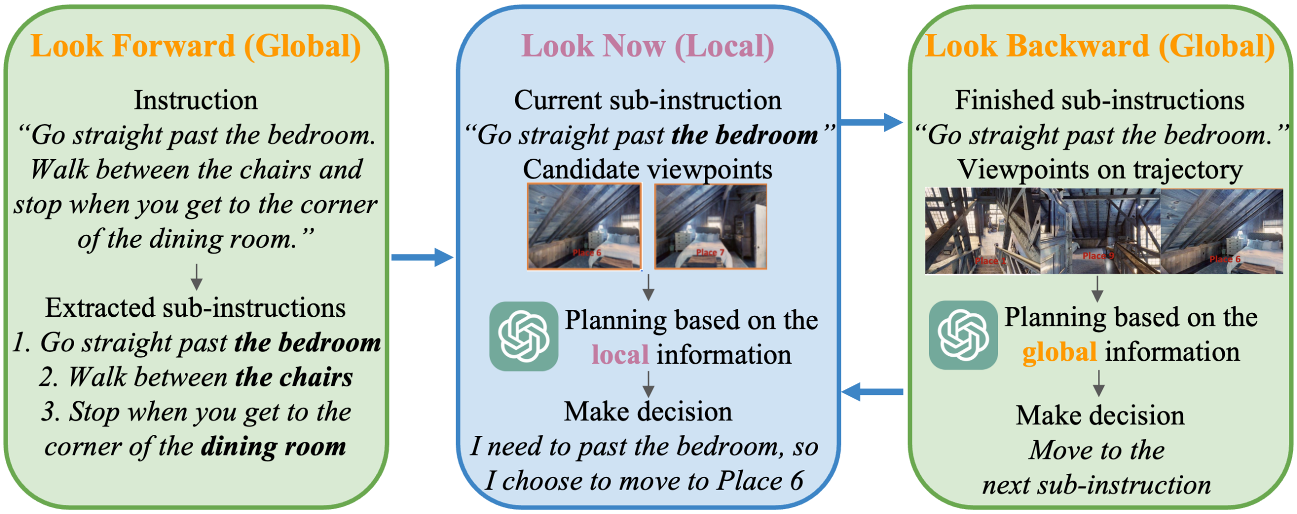

Figure 2: Illustration of the overall pipeline of the proposed methodology. We have three modules: *look forward*, *look now*, and *look backward*. The agent first looks forward to decomposing the natural-language instruction into an ordered list of sub-instructions and to extracting salient global landmarks that sketch a coarse route. Next, it looks now by matching the current observation against the active sub-instruction, and selecting the next waypoint for fine-grained local progress. Finally, it looks backward to audit the trajectory completed so far—revisiting stored viewpoints, verifying that finished sub-instructions were indeed satisfied, and triggering corrective backtracking if drift is detected.

bilities, sparking significant interest in their application to navigation tasks. Zhou et al. (2024b) introduced NavGPT, a purely LLM-based navigation agent that performs zero-shot sequential action prediction in VLN tasks by utilizing textual descriptions of visual observations, navigation history, and future explorable directions. Building upon this, a subsequent work Zhou et al. (2024a) aimed to bridge the gap between LLM-based agents and VLN-specialized models by aligning visual content within a frozen LLM and incorporating navigation policy networks. Navid (Zhang et al., 2024) adapted Vicuna-based LLMs to embodied navigation, while MapGPT (Chen et al., 2024) and DiscussNav (Long et al., 2024) explored topological textual map-guided exploration and multi-expert collaboration. These prior studies have largely adopted a step-by-step navigation paradigm that leverages textual memory to maintain long-term context, but such designs are prone to accumulated drift and may suffer from inaccuracies in progress estimation. In contrast, our method introduces explicit global–local reasoning with trajectory-level verification, providing a more structured understanding of navigation.

MLLM-Core Planner in VLN-CE Early work on VLN assumed discrete graph navigation, but the VLN-CE formulation of Krantz et al. (2020) exposed the far tougher problem of planning in photorealistic continuous spaces, where agents must issue velocity commands while coping with long horizons, compounding pose error, and partial observability. Open-Nav (Qiao et al., 2025) addressed this by coupling constraint reasoning with backtracking to enhance zero-shot robustness, while CA-Nav (Chen et al., 2025b) cast zero-shot VLN-CE as sequential constraint satisfaction to ensure both language and spatial criteria are met. More recently, SmartWay (Shi et al., 2025) integrated waypoint prediction with trajectory-level consistency checks. Different from these approaches, our framework not only enables MLLMs to conduct semantic analysis of the current environment for local navigation among candidate directions, but also equips them with meta-abilities to support higher-level decision making and a deeper understanding of the navigation process.

3 Method

3.1 Problem Formulation

Vision-and-Language Navigation in Continuous Environments (VLN-CE) is defined by an autonomous agent navigating within a continuous 3D environment $\mathcal{E}$ to reach a specified goal location based on natural language instructions. Let x_t denote the agent's pose (position and orientation) at time t. At each time step, the agent receives an observation $o_t \in \mathcal{O}$ consisting of a panoramic RGB-D view from its current pose (e.g.,

Figure 3: One successful example in the R2R-CE dataset. (a) *Look now*. Display the current candidate viewpoints and their short descriptions; the MLLM selects the option most aligned with the active sub-goal s_k as the next direction $v_t^\star$. (b) *Look backward*. Visualize all previously chosen viewpoints to form the executed trajectory τ; the MLLM audits τ against the finished sub-instructions to decide whether the completed sub-goals is satisfied.

12 RGB images $I_{t,1}^{\mathrm{RGB}}, \ldots, I_{t,12}^{\mathrm{RGB}}$ and 12 depth images $I_{t,1}^{\mathrm{D}}, \ldots, I_{t,12}^{\mathrm{D}}$ captured at 30° intervals for a full 360° panorama). The agent is also given a natural language instruction $W = (w_1, w_2, \ldots, w_L)$, where each w_i is a word token and L is the instruction length, describing the instructions to reach the goal for the agent. The action space $\mathcal{A}$ comprises discrete low-level navigation actions (e.g., turning or moving forward by fixed increments) that change the agent's pose. Starting from an initial pose x_0, the agent iteratively selects an action $a_t \in \mathcal{A}$ based on the instruction W and current observation o_t, yielding a new state x_{t+1} and observation o_{t+1} at the next step. This perception–action loop repeats until the agent executes a stop action upon reaching the destination. By following the instruction in this sequential decision process, the agent aims to minimize navigation errors and to successfully arrive at the target location in $\mathcal{E}$.

3.2 Overview

As shown in Fig. 2, our methodology consists of a lightweight three-view planner, dubbed Three-Step Nav, that augments any frozen multimodal LLM through prompt engineering. Previous Local Chain-of-Thought (CoT) approaches, which make decisions solely based on the current observation, are limited to reasoning over candidate viewpoints and a textual history memory. As a result, they are prone to drifting or becoming distracted over long trajectories. To address this issue, we introduce a **Hierarchical Global–Local Planner** that organizes navigation into three complementary stages: *look forward*, *look now*, and *look backward*. In the *look forward* stage, the instruction is decomposed into an ordered sequence of sub-instructions, each anchored by salient global landmarks, which together form a coarse-grained plan. Navigation then proceeds in a loop over these sub-goals. For each sub-goal, the agent repeatedly performs *look now*, selecting from candidate viewpoints the option most aligned with the current sub-instruction and its associated landmark. This local decision-making is guided by the MLLM navigator, which also checks whether the agent has approached the target landmark closely enough to consider the sub-goal completed. Once a sub-goal is tentatively reached, the agent executes *look backward*, auditing its past trajectory by replaying visited viewpoints and verifying that completed sub-instructions and landmarks were indeed satisfied from a global perspective. Only if the MLLM navigator confirms consistency with the history does the planner advance to the next sub-goal; otherwise, it triggers backtracking or refinement. This hierarchical global–local structure enables the agent to maintain long-horizon coherence while still making fine-grained, adaptive local decisions, effectively reducing premature stops and accumulated drift in zero-shot navigation. The methods are summarized in sections 3.3, 3.4, and 3.5 below.

3.3 Look forward

The looking-forward module converts the full natural-language instructions into an ordered "road-map" of sub-instructions and coarse global waypoints before any physical movement begins. Prompted with the instruction text and the agent's initial panoramic view, the frozen multimodal LLM (i) segments the instruction wherever spatial connectives (e.g., "past," "between," "until") or punctuation mark distinct goals, producing a sequence of atomic sub-instructions, and (ii) highlights the salient nouns that refer to persistent landmarks—rooms, furniture, or objects—that can anchor long-range navigation. Each sub-instruction is then paired with its corresponding landmark, yielding a high-level plan that sketches a straight-line route through the environment. Because this step is purely prompt-based and training-free, it can be injected into any VLN pipeline with negligible overhead while providing the global context that subsequent modules exploit for local decision making and drift correction.

3.4 Look now

The looking-now module (Fig. 3 (a)) is the agent's fine-grained decision engine at each time step. Given the current sub-instruction s_k (provided by the look-forward module) and the agent's observation o_t at pose x_t, the agent must decide where to move next. We first enumerate a set of navigable candidate viewpoints $\mathcal{V}t = v_{t,1}, v_{t,2}, \ldots, v_{t,m}$ reachable from x_t (e.g. adjacent viewpoints returned by the simulator). Followed by prior work (Qiao et al., 2025), we use a transformer-based model to serve as the Waypoint Prediction module and take panoramic RGB and depth images to pinpoint potential navigation waypoints. For each candidate $v_{t,i} \in \mathcal{V}t$ generated by the Waypoint Predictor, we select a viewpoint image of the view in that direction and the corresponding description. Aligning the active sub-instruction with the present visual observation of the potential navigation waypoints, and select a fine-grained waypoint for local progress. We then prompt the multimodal language model with a query that includes (i) the active sub-instruction s_k, (ii) a short description of the candidate viewpoints and the images (iii) the textual description for the history movement. The prompt asks MLLM to judge which viewpoint viewpoint will meaningfully advance the sub-goal described by s_k. The agent then selects the top-scoring direction v_t as the next waypoint to pursue. The low-level motion command corresponding to v_t is executed, causing the agent to navigate from x_t to the new location x_{t+1} and yielding a new observation o_{t+1}. In addition, the MLLM is required to output an estimated distance d_t to the landmarks mentioned in s_k; if d_t falls below a predefined threshold, the current sub-goal is considered ready to be inspected.

3.5 Look backward

As shown in Fig. 3 (b), the look backward module provides trajectory-level verification to catch cumulative drift before the agent terminates. After each sub-instruction—or whenever the agent believes the goal is reached—it assembles a compact textual replay of the visited viewpoints: a chronologically ordered list of landmark names, object mentions, and distances traveled. This replay, together with the finished sub-instructions, is fed back to the frozen MLLM with a prompt inspect two questions: (i) whether the current trajectory satisfies each sub-instruction in order and (ii) whether any overlooked landmark or missed turn suggests a correction. After generate the answer about these questions, the agent should invoke invoke one of four meta-abilities:

continue. If the distances and audit confirm the current sub-goal is satisfied, advance to s_{k+1}.

stay. If signals are borderline or uncertain, remain at x_t and re-query the MLLM without changing the trajectory.

backtrack. If the audit fails, roll back to the last reliable waypoint x_r and truncate the trajectory $\tau \leftarrow \tau_{0:r}$.

look-around. If uncertainty is high, temporarily visit all candidate neighbor viewpoints $v \in \mathcal{N}(x_t)$ to collect observations from these neighbor nodes, then return to x_t for re-evaluation.

By closing this audit loop at runtime—without gradient updates or environment-specific heuristics—looking backward markedly reduces premature stops and large navigation errors, ensuring that global intent aligns with the final executed route.

4 Experiments

4.1 Experiment Setup

Dataset We conduct our framework on two standard benchmarks for vision-and-language navigation in continuous environments: (Ku et al., 2020) and RxR-CE (Krantz et al., 2020). R2R-CE extends the Room-to-Room dataset (Anderson et al., 2018) to continuous settings based on the Habitat simulator (Savva et al., 2019). Compared to R2R-CE, RxR-CE introduces longer instructions, longer paths, stricter physi-

Table 1: Comparison with supervised and zero-shot methods on validation unseen split of R2R-CE. **Bold** denotes the best performance across all zero-shot methods, while Underlined indicates the second-best results across zero-shot methods.

Method	TL	NE↓	nDTW↑	OSR↑	SR↑	SPL↑
Supervised Learning						
CMA (Hong et al., 2022)	11.08	6.92	50.77	45	37	32.17
RecBERT (Hong et al., 2022)	11.06	5.80	54.81	57	48	43.22
BEVBert (An et al., 2023)	13.63	5.13	61.40	64	60	53.41
Navid (Zhang et al., 2024)	7.63	5.47	–	49	37	35.90
ETPNav (An et al., 2024)	11.08	5.15	61.15	58	52	52.18
Zero-Shot						
Random	8.15	8.63	34.08	12	2	1.50
LXMERT (Hong et al., 2022)	15.79	10.48	18.73	22	2	1.87
DiscussNav (Long et al., 2024)	6.27	7.77	42.87	15	11	10.51
MapGPT-CE (Chen et al., 2024)	12.63	8.16	–	21	7	5.04
NavGPT-CE (Zhou et al., 2024b)	–	8.37	–	27	16	10.20
Open-Nav (Qiao et al., 2025)	7.68	6.70	45.79	23	19	16.10
AO-Planner (Chen et al., 2025a)	–	6.95	–	38	25	16.60
CA-Nav (Chen et al., 2025b)	–	7.58	–	48	25	10.80
SmartWay (Shi et al., 2025)	13.09	7.01	–	**51**	29	22.46
Ours	9.18	**5.87**	**57.70**	39	**34**	**29.12**

Table 2: Comparison with supervised and zero-shot methods on validation unseen split of RxR-CE.

Method	NE↓	nDTW↑	SR↑	SPL↑
Supervised Learning				
Seq2Seq Krantz et al. (2020)	12.10	30.8	13.9	11.9
DC-VLN Hong et al. (2022)	8.98	46.7	27.1	22.7
Navid Zhang et al. (2024)	8.41	–	23.8	21.2
ETPNav An et al. (2024)	5.64	61.9	54.8	44.9
Zero-Shot				
CLIP-Nav (Dorbala et al., 2022)	–	–	9.8	3.2
A2Nav Chen et al. (2023)	–	–	16.8	6.3
AO-Planner Chen et al. (2025a)	10.75	33.1	**22.4**	15.1
CA-Nav Chen et al. (2025b)	10.37	13.5	19.0	6.0
Ours	**9.21**	**45.7**	22.0	**16.1**

cal restrictions, and greater risk of getting stuck, making it significantly more challenging. Following the setting in previous work (Qiao et al., 2025), we use the same 100 selected episodes from the val-unseen validation splits of R2R-CE, and randomly sampled 100 episodes from the English val-unseen split of RxR-CE, to balance coverage and API efficiency. For each episode, the agent receives a natural language instruction and must reach the corresponding goal in the Habitat simulator, with the gpt-5-2025-08-07 API serving as the core of the multimodal planner.

Evaluation Metrics We utilize the pre-defined evaluation metrics of the VLN task. (1) Trajectory Length (TL), the total distance traveled by the agent; (2) Navigation Error (NE), the shortest geodesic distance between the final position and the goal; (3)

Table 3: Ablation study on the validation unseen split of R2R-CE. We compare the full Hierarchical Global–Local Planner with ablated variants that disable global reasoning modules.

Method	Step1	Step2	Step3	NE↓	nDTW↑	SR↑
Ours	✓	✓	✓	**5.87**	**57.70**	**34**
Ours (only local view)	–	✓	–	6.55	54.52	20
Ours (w/o Look Backward)	✓	✓	–	6.22	55.85	28

Normalized Dynamic Time Warping (nDTW), trajectory similarity by aligning the executed and reference paths; (4) Success Rate (SR), the percentage of episodes where the agent stops within 3 meters of the goal; (5) Object Success Rate (OSR), Success rate of reaching the goal along the trajectory; (6) Success weighted by Path Length (SPL), weighting successful trajectories according to the ratio of the shortest-path distance to the actual path length.

4.2 Experimental Results

Table 1 shows a comprehensive performance comparison of both supervised learning and zero-shot methods on the val-unseen split of R2R-CE. Supervised models, trained with extensive domain-specific data, naturally obtain higher success rates (up to 52–60%) and maintain strong results across all metrics, highlighting the advantage of task-tailored learning in controlled settings. In contrast, zero-shot agents are deployed without task-specific fine-tuning and generally achieve lower SPL and SR, reflecting the inherent diffi-

Table 4: Comparison of different multimodal LLM experts on the validation unseen split of R2R-CE. We report results using our full Hierarchical Global–Local Planner.

MLLM Expert	NE↓	nDTW↑	SR↑
GPT-5	**5.87**	**57.70**	**34**
GPT-4o	6.70	56.42	28
GPT-4v	6.94	54.10	26

culty of transferring general reasoning abilities to embodied navigation. Within this challenging zero-shot regime, our Three-Step Nav achieves the best overall performance, reaching an SR of 34%, an NE of 5.87, and an SPL of 29.12%. Compared with the strongest prior zero-shot SOTA methods, this corresponds to an improvement of 5% in SR and 6.66% in SPL, meanwhile we gain a relatively reduction of 12.4% in NE. These improvements indicate that our Hierarchical Global-Local Planner framework enables more decision-making and avoids unnecessary navigation error, resulting in stronger goal-reaching capability and trajectory efficiency. Overall, the results demonstrate that incorporating structured global–local reasoning into an MLLM-based planner significantly boosts zero-shot performance and narrows the gap with fully supervised navigation systems.

Table 2 reports the results on the RxR-CE dataset. Supervised approaches such as ETPNav, which are trained with extensive task-specific data, achieve strong performance (e.g., SR of 54.8% and SPL of 44.9), but require large-scale supervision that limits generalizability. In contrast, zero-shot models operate without fine-tuning and generally lag behind in SPL and SR. Within this challenging regime, our Three-Step Nav achieves competitive results, with an NE of 9.21, an nDTW of 45.7, an SR of 22.0, and an SPL of 16.1. Compared with the strongest zero-shot baselines, our method improves nDTW by 12.6%, and relatively reduces NE by 11.2%. These results highlight that the integration of global–local reasoning and trajectory-level auditing enables robust progress estimation in long-horizon navigation, even under the more demanding RxR-CE benchmark.

4.3 Ablation Study

Table 3 reports the ablation results on R2R-CE, highlighting the contributions of different modules in our hierarchical design. The full model with all three steps enabled achieves the strongest performance (NE 5.87, nDTW 57.70, SR 34%). When only the local reasoning module (*look now*) is active, performance drops substantially (SR 20%, NE 6.55), showing that local chain-of-thought reasoning alone cannot maintain long-range consistency. Enabling global forward planning but removing the backward verification improves results to SR 28% and nDTW 55.85, yet still lags behind the full framework. These comparisons demonstrate that both global planning (*look forward*) and trajectory-level auditing (*look backward*) are indispensable: the forward step provides high-level guidance, while the backward check prevents drift and premature stops. Together, they enable our Three-Step Nav to achieve robust zero-shot navigation beyond local-only baselines.

To further understand the role of the underlying MLLM expert, Table 4 compares GPT-5, GPT-4o, and GPT-4V within our full Three-Step Nav framework. Among the three tested models, GPT-5 delivers the strongest results (NE 5.87, nDTW 57.70, SR 34%), demonstrating its superior capability for long-horizon spatial reasoning. Using GPT-4o still provides competitive performance, but with a noticeable drop in SR (28%) and slightly higher navigation error (NE 6.70), while GPT-4v lags further behind with SR 26% and NE 6.94. These comparisons suggest that although our planner consistently benefits from structured global–local reasoning regardless of the underlying expert, more advanced MLLMs such as GPT-5 yield significant gains in both success rate and trajectory fidelity. This highlights the importance of model quality when scaling zero-shot navigation frameworks to more complex environments.

5 Conclusions

Summary In this work, we introduced Three-Step Nav, a hierarchical global–local planner that enables zero-shot VLN in continuous environments without any task-specific training. Our three-stage approach – look forward, look now, and look backward – allows a frozen multimodal LLM to maintain long-horizon context, make fine-grained decisions, and self-correct by backtracking. This framework is lightweight and modular, easily integrating into existing navigation pipelines. Three-Step Nav set a new state-of-the-art among zero-shot methods on R2R-CE, reducing navigation error by about 15% and substantially improving success rate and path efficiency (SPL) compared to previous approaches. It also achieved strong performance on the challenging RxR-CE benchmark, demonstrating competitive results close to fully supervised policies. These gains validate that coupling global route planning with trajectory-level verification can significantly mitigate distraction and drift in long-horizon navigation.

Method Assumptions and Limitations While effective, our approach assumes access to a powerful multimodal language model and a relatively static environment. The agent's reasoning heavily relies on the pre-trained LLM (e.g. GPT-5), which introduces computational cost and potential errors if the model misinterprets visual cues or instructions. Also, our evaluations are in simulated indoor environments – transferring to real robots will require handling sensor noise, moving obstacles, and other real-world complexities. The Three-Step Nav framework does not learn from feedback over time, so extremely long or ambiguous instructions can still pose challenges. In future work, we plan to refine the system's robustness by incorporating learning-based adaptation or fine-tuning smaller navigation-specific models. Additionally, optimizing the prompting strategy and LLM integration could reduce latency, making the system more feasible for real-time robotic deployment. Validating our planner on physical robots in real homes or public spaces is an important next step toward closing the sim-to-real gap.

Societal Impact Improving Vision-and-Language Navigation has positive implications for assistive robotics and autonomous agents in society. A robust zero-shot navigation agent could assist visually impaired users in unfamiliar indoor spaces or enable home robots to follow complex spoken instructions, enhancing accessibility and convenience. Our method's ability to self-audit and correct mistakes is particularly valuable for safety-critical applications like search-and-rescue, where backtracking from a wrong turn can prevent accidents. We are mindful that any autonomous navigation system must be thoroughly tested to avoid hazards – for instance, a navigation error in a real home could lead to collisions. By reducing drift and improving reliability, Three-Step Nav contributes to safer deployment of embodied AI. Continued research should examine ethical considerations (such as minimizing biases in language understanding) and include stakeholders in testing to ensure these systems benefit users of diverse backgrounds. Overall, our hierarchical planner represents a step toward more trustworthy and general-purpose embodied agents, bridging the gap between simulation benchmarks and real-world navigation tasks.

References

An, D., Qi, Y., Li, Y., Huang, Y., Wang, L., Tan, T., and Shao, J. (2023). Bevbert: Multimodal map pre-training for language-guided navigation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

An, D., Wang, H., Wang, W., Wang, Z., Huang, Y., He, K., and Wang, L. (2024). Etpnav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., and Van Den Hengel, A. (2018). Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683.

Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., and Zhang, Y. (2017). Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*.

Chen, J., Lin, B., Liu, X., Ma, L., Liang, X., and Wong, K.-Y. K. (2025a). Affordances-oriented planning using foundation models for continuous vision-language navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23568–23576.

Chen, J., Lin, B., Xu, R., Chai, Z., Liang, X., and Wong, K.-Y. (2024). Mapgpt: Map-guided prompting with adaptive path planning for vision-and-language navigation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9796–9810.

Chen, K., An, D., Huang, Y., Xu, R., Su, Y., Ling, Y., Reid, I., and Wang, L. (2025b). Constraint-aware zero-shot vision-language navigation in continuous environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–16.

Chen, P., Sun, X., Zhi, H., Zeng, R., Li, T. H., Liu, G., Tan, M., and Gan, C. (2023). A^{2}nav: Action-aware zero-shot robot navigation by exploiting vision-and-language ability of foundation models. *arXiv preprint arXiv:2308.07997*.

Chen, S., Guhur, P.-L., Tapaswi, M., Schmid, C., and Laptev, I. (2022). Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16537–16547.

Dorbala, V. S., Sigurdsson, G., Piramuthu, R., Thomason, J., and Sukhatme, G. S. (2022). Clip-nav: Using clip for zero-shot vision-and-language navigation.

Gu, J., Stefani, E., Wu, Q., Thomason, J., and Wang, X. (2022). Vision-and-language navigation: A survey of tasks, methods, and future directions. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume*

1: Long Papers)*, pages 7606–7623, Dublin, Ireland. Association for Computational Linguistics.

Hong, Y., Wang, Z., Wu, Q., and Gould, S. (2022). Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15439–15449.

Krantz, J., Wijmans, E., Majumdar, A., Batra, D., and Lee, S. (2020). Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pages 104–120. Springer.

Ku, A., Anderson, P., Patel, R., Ie, E., and Baldridge, J. (2020). Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4392–4412.

Li, J., Padmakumar, A., Sukhatme, G., and Bansal, M. (2024). Vln-video: Utilizing driving videos for outdoor vision-and-language navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18517–18526.

Long, Y., Li, X., Cai, W., and Dong, H. (2024). Discuss before moving: Visual language navigation via multi-expert discussions. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 17380–17387. IEEE.

Qiao, Y., Lyu, W., Wang, H., Wang, Z., Li, Z., Zhang, Y., Tan, M., and Wu, Q. (2025). Open-nav: Exploring zero-shot vision-and-language navigation in continuous environment with open-source llms. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6710–6717. IEEE.

Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., et al. (2019). Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347.

Shi, X., Li, Z., Lyu, W., Xia, J., Dayoub, F., Qiao, Y., and Wu, Q. (2025). Smartway: Enhanced waypoint prediction and backtracking for zero-shot vision-and-language navigation.

Wang, Z., Li, J., Hong, Y., Wang, Y., Wu, Q., Bansal, M., Gould, S., Tan, H., and Qiao, Y. (2023a). Scaling data generation in vision-and-language navigation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12009–12020.

Wang, Z., Li, X., Yang, J., Liu, Y., and Jiang, S. (2023b). Gridmm: Grid memory map for vision-and-language navigation. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 15625–15636.

Zhang, J., Wang, K., Xu, R., Zhou, G., Hong, Y., Fang, X., Wu, Q., Zhang, Z., and Wang, H. (2024). Navid: Video-based vlm plans the next step for vision-and-language navigation. *Robotics: Science and Systems*.

Zhou, G., Hong, Y., Wang, Z., Wang, X. E., and Wu, Q. (2024a). Navgpt-2: Unleashing navigational reasoning capability for large vision-language models. In *European Conference on Computer Vision*, pages 260–278. Springer.

Zhou, G., Hong, Y., and Wu, Q. (2024b). Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7641–7649.

Zhu, F., Lee, V. C., Chang, X., and Liang, X. (2023). Vision language navigation with knowledge-driven environmental dreamer. In *International Joint Conference on Artificial Intelligence 2023*, pages 1840–1848. Association for the Advancement of Artificial Intelligence (AAAI).