

LaViRA: Language-Vision-Robot Actions Translation for Zero-Shot Vision Language Navigation in Continuous Environments

Hongyu Ding^{1,*}, Ziming Xu^{1,*}, YUK TUNG SAMUEL FANG², You Wu¹, Zixuan Chen¹,
Jieqi Shi^{2,†}, Jing Huo^{1,†}, Yifan Zhang³, Yang Gao²

Abstract—Zero-shot Vision-and-Language Navigation in Continuous Environments (VLN-CE) requires an agent to navigate unseen environments based on natural language instructions without any prior training. Current methods face a critical trade-off: either rely on environment-specific waypoint predictors that limit scene generalization, or underutilize the reasoning capabilities of large models during navigation. We introduce LaViRA, a *simple yet effective* zero-shot framework that addresses this dilemma by decomposing action into a coarse-to-fine hierarchy: *Language Action* for high-level planning, *Vision Action* for middle-level perceptual grounding, and *Robot Action* for low-level control. This modular decomposition allows us to leverage the distinct strengths of different scales of Multimodal Large Language Models (MLLMs) at each stage, creating a system that is powerful in its reasoning, grounding and practical control. LaViRA significantly outperforms existing state-of-the-art methods on the VLN-CE benchmark, demonstrating superior generalization capabilities in unseen environments, while maintaining transparency and efficiency for real-world deployment. Project page: <https://robo-lavira.github.io/lavira-zs-vln/>

I. INTRODUCTION

Vision-and-Language Navigation (VLN) presents the challenge of grounding natural language instructions within visual observations to enable an embodied agent to navigate through previously unseen environments [1]. Early VLN research was primarily conducted in discrete, graph-based settings where navigation is simplified to selecting paths between predefined nodes. To bridge the gap to the real world, Vision-and-Language Navigation in Continuous Environments (VLN-CE) [2] was introduced, removing the reliance on connectivity graphs and forcing agents to contend

*Equal Contribution, †Corresponding Author

¹Hongyu Ding, Ziming Xu, You Wu, Zixuan Chen and Jing Huo are with the School of Computer Science, Nanjing University, China. Emails: {hongyuding, zimingxu, you}@smail.nju.edu.cn, {chenzx, huojing}@nju.edu.cn

²YUK TUNG SAMUEL FANG, Jieqi Shi and Yang Gao are with the School of Intelligence Science and Technology, Nanjing University, China. Emails: 231880023@smail.nju.edu.cn, isjieqi@nju.edu.cn, gaoy@nju.edu.cn

³Yifan Zhang is with the Institute of Automation, Chinese Academy of Sciences, and the University of Chinese Academy of Sciences, Nanjing, China. Email: yfzhang@nlpr.ia.ac.cn

This work is supported in part by New Generation Artificial Intelligence-National Science and Technology Major Project (2025ZD0122904), National Natural Science Foundation of China (62506153, 62192783, 62276128, 62273347), Jiangsu Science and Technology Major Project (BG2025035), Fundamental Research Funds for the Central Universities (KG202514), Jiangsu Postgraduate Research & Practice Innovation Program (KYCX25_0300) and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

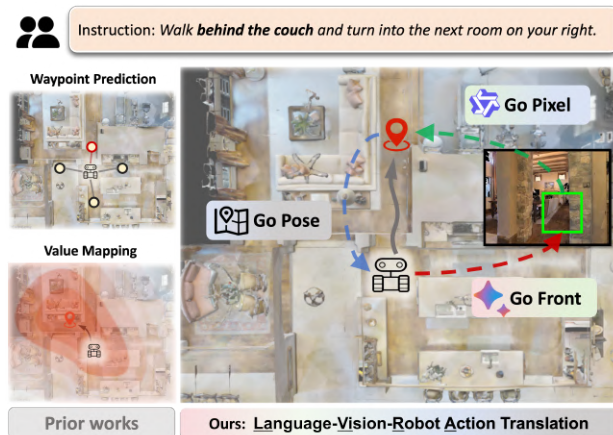


Fig. 1: Prior methods rely on pre-trained waypoint prediction or value mapping with limited online planning. Our LaViRA framework instead decomposes navigation into language-level planning (“Go Front”), vision-level grounding (“Go Pixel”), and robot-level control (“Go Pose”), fully leveraging MLLMs reasoning for coarse-to-fine decision-making.

with realistic challenges like continuous visual perception and fine-grained motor control.

The recent success of LLMs and MLLMs has inspired zero-shot VLN-CE [3]–[6], where agents navigate without environment-specific training. Two main paradigms have emerged. **Waypoint prediction with large model reasoning.** These methods combine a pre-trained waypoint predictor with an LLM or MLLM [3], [4]. The predictor proposes navigable waypoints, and the large model reasons over these candidates to select the next step. This leverages high-level planning but requires separate pre-training and struggles to generalize to unseen environments. **Value mapping with vision language models.** An alternative discards the waypoint predictor and uses a vision-language model (e.g., BLIP [7]) to generate a semantic relevance heatmap over the scene. The agent navigates toward the highest-scoring region. While this avoids predictor training, it typically limits powerful large models to offline instruction parsing, underutilizing their reasoning capabilities during online navigation.

These two paradigms reveal a fundamental trade-off. Waypoint-based methods excel at high-level reasoning but are constrained by a separate, inflexible waypoint prediction module. Value-mapping methods are more perceptually

grounded but lack dynamic, high-level reasoning during navigation. This leads us to a simple but critical question:

Can we design a purely zero-shot VLN-CE framework that (1) removes the dependency on a pre-trained waypoint predictor and (2) fully harnesses the reasoning abilities of MLLMs for navigation decision making?

We answer this question with **LaViRA**: Language-Vision-Robot Action Translation. Our key insight is to decompose navigation into a coarse-to-fine, multi-stage action space, progressively refined from language to vision to robot control. Grounded in the divide-and-conquer principle for long-horizon decision making under partial observability [8] and hierarchical reinforcement learning theory [9], this decomposition exponentially reduces the search space at each stage, improving decision efficiency and zero-shot generalization in unseen continuous environments. Rather than requiring a single model to produce low-level controls directly, we allocate each stage to a model scale that best matches its reasoning or perceptual demands, allowing the system to exploit the complementary strengths of different models.

- 1) **Language Action:** A powerful MLLM acts as a high-level planner, analyzing the instruction, history, and current observation to produce a coarse strategic decision, such as which general direction to head, whether to backtrack, or when to stop.
- 2) **Vision Action:** A smaller, efficient MLLM takes this high-level decision and grounds it in the visual scene, identifying a specific object or region to move towards.
- 3) **Robot Action:** A simple, rule-based controller executes the low-level movement to the identified target.

This hierarchical decomposition offers three key advantages. First, it is *purely zero-shot*, eliminating any need for pre-trained waypoint predictors. Second, it *fully engages MLLM reasoning* across multiple granularities: high-level planning, mid-level perceptual grounding, and low-level control. Third, its modular design provides *transparency and practicality*, allowing seamless adaptation to both simulated and real-world robots.

Our contributions are as follows:

- We propose a general action decomposition strategy for zero-shot VLN-CE that separates navigation into language-level planning, vision-level grounding, and robot-level control, enabling flexible integration of reasoning and perception modules.
- We instantiate this strategy in LaViRA, a practical Language-Vision-Robot Action framework that leverages multi-scale MLLMs in a fully zero-shot manner.
- We achieve state-of-the-art zero-shot performance on the VLN-CE benchmark while maintaining strong effectiveness and efficiency for real-world deployment.

II. RELATED WORK

A. Vision-and-Language Navigation

Early research in Vision-and-Language Navigation (VLN) [1], where an agent follows instructions to navigate

unseen environments, predominantly focused on discrete, graph-based settings [10]–[12]. Such environments enable high-level decision-making but fail to capture the continuous control demands of real-world scenarios. To address this, VLN in Continuous Environments (VLN-CE) [2] removes reliance on connectivity graphs and requires agents to perform fine-grained control actions like moving forward, rotating, and avoiding obstacles. This transition introduces new challenges in scene analysis, generalization, and low-level control.

Numerous VLN methods in both discrete and continuous environments have improved performance through enhanced cross-modal alignment [13], [14], reinforcement learning [15], data augmentation [16]–[19], and map-based representations [20]–[22]. However, these learning-based approaches require substantial environment-specific training, which limits their applicability for zero-shot deployment—motivating recent interest in training-free VLN-CE solutions.

B. Zero-Shot VLN with Foundation Models

The rise of foundation models such as LLMs [23], [24] and MLLMs [7], [25], [26] has driven a new wave of zero-shot VLN-CE research, categorized by how they integrate with robot control. One dominant paradigm is *waypoint-based navigation* [3], [4]. These methods use an LLM/MLLM to select waypoints proposed by a pre-trained predictor [27]. However, this creates a critical dependency on the predictor, which often fails to generalize to unseen scenes and limits backtracking flexibility [4]. Another approach is *heuristic value-mapping* [5], [6], where a Vision-Language Model (VLM) generates a semantic heatmap to guide the agent. In these frameworks, powerful LLMs are restricted to offline instruction parsing, underutilizing their dynamic reasoning during navigation. Reliance on hard constraints for progress estimation also introduces rigidity in complex scenarios [5]. Recently, GC-VLN [28] models instructions as explicit graph constraints for training-free navigation. While effective for structured reasoning, it still requires offline parsing and lacks flexible backtracking.

Our LaViRA framework bridges the gap between these approaches, introducing a *Language-Vision-Robot Action Translation* that progressively refines action from language-level planning to vision-level grounding to robot-level execution. This design allows MLLMs reasoning to directly influence decisions at multiple granularities, enabling fully zero-shot, interpretable, and adaptable navigation in continuous environments.

C. Hierarchical Structure in VLN

Hierarchical architectures have emerged as a powerful paradigm for addressing complex, long-horizon navigation tasks. This strategy effectively decomposes the problem into high-level planning and low-level execution, allowing different modules to specialize. Current hierarchical approaches often combine high-level semantic planning with low-level motion control. One common strategy involves dividing the

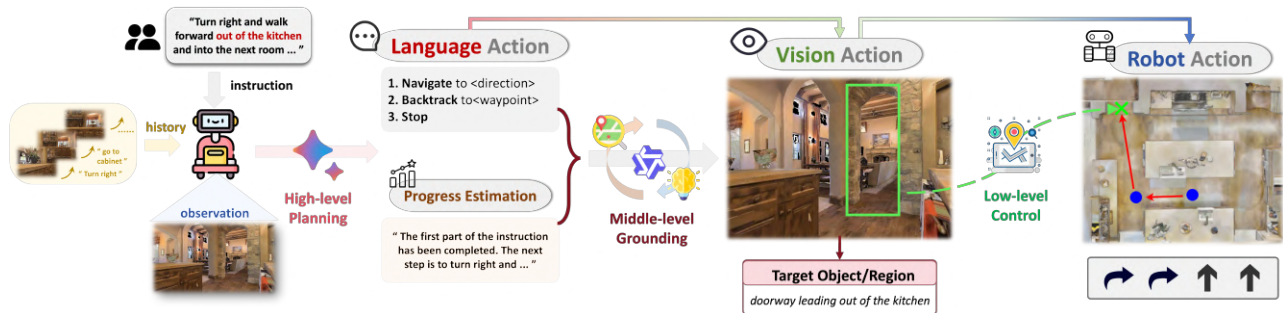


Fig. 2: **The LaViRA Pipeline.** Our framework decomposes navigation into three sequential stages. (1) **Language Action:** A large MLLM processes the instruction, history, and current observation to generate a high-level plan, deciding whether to move forward, turn, backtrack, or stop. (2) **Vision Action:** A smaller, more efficient MLLM takes the high-level plan and progress estimation to identify a specific visual target in the chosen direction, outputting its bounding box and description. (3) **Robot Action:** The target’s pixel coordinates are projected onto a global map, and a rule-based controller navigates the robot to the destination. This hierarchical process enables generalized and robust zero-shot vision-language navigation.

environment into distinct topological regions and having a high-level planner select a sequence of zones to traverse, while a low-level controller navigates within each zone [29], [30]. Other methods learn policies for both the manager and worker through techniques like imitation learning [31] or reinforcement learning [32]. A recent example, *NavA*³ [33], uses a global policy to identify target regions and a pre-trained model for fine-grained navigation.

However, a common limitation of these methods is their reliance on extensive, environment-specific training, which restricts generalization and hinders zero-shot deployment. In contrast, LaViRA’s Language-Vision-Robot action decomposition is fully zero-shot. By leveraging MLLM reasoning and a rule-based controller, it eliminates the need for any training, which simplifies deployment and enhances modularity.

III. PROPOSED METHOD

Our approach, LaViRA, introduces a novel framework for zero-shot Vision-and-Language Navigation in Continuous Environments (VLN-CE). The core idea is simple yet effective: we decompose the complex navigation task into a coarse-to-fine hierarchy of actions: *Language Action*, *Vision Action*, and *Robot Action*. This modular decomposition allows us to leverage the distinct strengths of different scales of Multimodal Large Language Models (MLLMs) at each stage, creating a system that is powerful in its reasoning, grounding and practical for real-world deployment, all without requiring any environment-specific training.

The overall pipeline is illustrated in Figure 2. In the VLN-CE task, an agent must follow a natural language instruction \mathcal{I} through an unseen environment. At each timestep t , it uses an egocentric observation I_t to choose its next action \mathcal{A}_t from a continuous space. To address this, our method decomposes the navigation process into a sequence of three hierarchical actions: a high-level directional plan (Language Action), the grounding of this plan into a specific visual target (Vision Action), and finally, the low-level movement to reach it (Robot Action). We detail each stage below.

A. Language Action: High-Level Planning

The first stage of our framework addresses the question: *Where should I generally go next?* To answer this, we employ a powerful, large-scale MLLM (e.g., GPT-4o) that functions as a high-level planner. This model is responsible for interpreting the full context of the navigation task and producing a coarse, directional command.

Specifically, the model receives three types of input:

- **Language Instruction \mathcal{I} :** The given natural language instruction provided at the start of the task.
- **Current Observation \mathcal{O}_t :** A set of four images corresponding to the agent’s front, left, right, and back views $\{I_{front}, I_{left}, I_{right}, I_{back}\}$, providing an informative understanding of the current location.
- **Navigation History \mathcal{H}_t :** A structured summary of past observations and actions, formatted as a sequence like $\mathcal{H}_t = \{(\mathcal{O}_0, \mathcal{A}_0), (\mathcal{O}_1, \mathcal{A}_1), \dots, (\mathcal{O}_{t-1}, \mathcal{A}_{t-1})\}$. This provides crucial context on what has already been accomplished.

Given this rich multimodal context, the MLLM performs two tasks simultaneously. First, it generates a **Progress Estimation \mathcal{P}_t** , a natural language assessment of how much of the instruction has been completed. This explicit reasoning step forces the model to track its progress against the overall instruction. Second, based on its analysis, the model selects a **Language Action \mathcal{A}_t^{lang}** from a discrete set:

- `navigate to <direction>`: Move forward, left, right, or behind.
- `backtrack to <waypoint>`: Return to a previously visited waypoint location.
- `stop`: Terminate the navigation.

This process can be formulated as:

$$(\mathcal{A}_t^{lang}, \mathcal{P}_t) = \text{MLLM}_{large}(\mathcal{I}, \mathcal{H}_t, \mathcal{O}_t) \quad (1)$$

This stage effectively abstracts the continuous environment into a few high-level choices, allowing the large MLLM to focus its powerful reasoning capabilities on strategic, long-term planning.

Language Action Prompt

You are an expert specialized in vision-language navigation task. Please review the contexts below and help the robot navigate.

Navigation Instruction

"Turn right and walk to the cabinet in front of you. Turn right and ..."

Current Observation

{ "front": , "left": , "back": , "right": }

Navigation History

"You start from , turn [right] and see , then go to the [cabinet] and see ,"

Action Options

1. **navigate to** [direction_id] - choose a direction and go forward
2. **backtrack to** [waypoint_id] - return to a previous waypoint
3. **stop** - end of navigation trajectory

Response format (JSON)

```
{
  "progress_analysis": "<assessment of current progress toward instruction completion>",
  "reasoning": "<explanation of chosen action>",
  "action": "navigate to forward/left/right/behind or "backtrack to [waypoint_id]" or "stop"
}
```

Guidelines

- Consider instruction completion progress
- Backtrack only if current path seems unproductive or dead-end
- Choose direction that best advances toward goal

Now, decide the **Language Action** based on the above contexts.

Vision Action Prompt

You are an expert specialized in vision-language navigation task. Please review the contexts below and help the robot navigate.

Navigation Instruction

"Turn right and walk to the cabinet in front of you. Turn right and ..."

Current Observation

{ "chosen direction image": }

Navigation History

- **Previously visited targets:** cabinet in the kitchen, doorway leading out of the kitchen,
- **Progress Analysis:** "The first part of the instruction has been completed. The next step is ..."

Your Task

1. Analyze at what stage the current instruction has been completed and what should be done next.
2. Identify the most relevant target object/region for what you should do next.
3. Specify **ONLY ONE** target object/region. And it should not be too close to you.

Response format (JSON)

```
{
  "progress": "<assessment of how close to completing the instruction>",
  "reasoning": "<brief explanation of decision>",
  "bbox_2d": [x1, y1, x2, y2],
  "target": "<description of target object/region>"
}
```

Guidelines

- Target description should be specific and clear
- Consider the instruction completion progress based on visited targets
- Use the progress analysis to inform your decision

Now, decide the **Vision Action** based on the above contexts.

Fig. 3: **Prompts for Language and Vision Actions.** (Left) The prompt for the Language Action model, which takes in full context to decide on a high-level direction. (Right) The prompt for the Vision Action model, which uses the output from the first stage to ground the decision in a specific visual target.

B. Vision Action: Perceptual Grounding

Once a high-level Language Action is determined, the next stage must answer: *What specific thing should I move towards in that direction?* This is the role of the Vision Action stage, where we ground the abstract plan into a concrete perceptual target.

For this task, we use a smaller, more efficient MLLM (e.g., Qwen2.5-VL-32B). This choice is deliberate: grounding is a more focused perception task that does not require the same extensive world knowledge as high-level planning, making a smaller model more suitable and computationally efficient. As our ablation studies in Section IV-C confirm, pairing a powerful planner with a specialized, efficient grounding model yields optimal performance. The model is prompted with:

- **Language Instruction** \mathcal{I} : The original instruction.
- **Progress Estimation** \mathcal{P}_t : The text generated by the Language Action model.
- **Chosen Direction Image** I_{dir} : The single image corresponding to the direction chosen in the previous stage \mathcal{A}_t^{lang} .

The model's task is to identify the most relevant object or region in the image that aligns with the next step of the instruction. It outputs a **Vision Action** \mathcal{A}_t^{vis} in a structured format containing a bounding box and its description. This can be expressed as:

$$\mathcal{A}_t^{vis} = \text{MLLM}_{small}(\mathcal{I}, \mathcal{P}_t, I_{dir}) \quad (2)$$

The output \mathcal{A}_t^{vis} is a dictionary containing:

- **Bounding Box** $bbox_{2d}$: The 2D coordinates $[x_1, y_1, x_2, y_2]$ localizing the target.
- **Target Description**: A textual description of the identified object or region.

As shown in Figure 3, the prompt instructs the model to select a target that is not too close, encouraging meaningful progress. This stage effectively translates the high-level plan into a tangible, visually verifiable goal.

C. Robot Action: Low-Level Control

The final Robot Action stage answers: *How do I physically get there?* It translates the identified visual target into low-level motor commands using a robust, rule-based controller. **Pixel-to-World Projection.** First, we select the bottom-center pixel of the target's bounding box and project it into a 3D point in the world frame. This involves unprojecting the 2D pixel to the camera's 3D coordinate system using the intrinsic matrix \mathbf{K} :

$$[X_{cam}, Y_{cam}, Z_{cam}]^T = d \cdot \mathbf{K}^{-1} \cdot [u_{target}, v_{target}, 1]^T \quad (3)$$

where d is the depth. This 3D point is then transformed from the camera frame to the world frame using the agent's current pose $\mathbf{T}_{agent} = (x_{agent}, z_{agent}, \theta_{agent})$ to yield a target position $\mathbf{p}_{world} = (x_{world}, z_{world})$ on the 2D map:

$$\begin{bmatrix} x_{world} \\ z_{world} \end{bmatrix} = \begin{bmatrix} x_{agent} \\ z_{agent} \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} Z_{cam} \\ -X_{cam} \end{bmatrix} \quad (4)$$

Path Planning and Control. Given the target position \mathbf{p}_{world} , the agent computes a short-term path using the Fast Marching Method (FMM). A low-level controller then executes this path with local obstacle avoidance. This deterministic final step grounds the reasoning chain in physical action, ensuring interpretability and making the system adaptable to different robot platforms by simply swapping the controller.

IV. SIMULATION EXPERIMENTS

We validate LaViRA in simulation to answer three key questions: (1) How does it compare against state-of-the-art methods on the standard VLN-CE benchmark? (2) Does performance support our hypothesis of using different MLLM scales for different decision granularities? (3) What is the contribution of each component in our framework?

A. Experimental Setup

Environment and Dataset. We use the Habitat simulator [34] with the VLN-CE dataset [2], which extends the R2R benchmark from Matterport3D (MP3D) [10] for continuous navigation. Following recent zero-shot works [3], [4], we report results on a standard 100-episode subset from the validation unseen split. An episode is successful if the agent stops within 3 meters of the target.

Evaluation Metrics. We use standard VLN metrics: *Navigation Error (NE)*, the final distance to goal; *Success Rate (SR)*, our primary metric for stopping within 3m; *Oracle Success Rate (OSR)*, SR if stopping at the closest point on the path; and *Success rate weighted by Path Length (SPL)*, which penalizes inefficient paths. To account for the *inherent stochasticity* in the outputs of MLLMs, we **repeat 3 runs** for each experiment over the 100-episode set and report the *mean and standard deviation* for all metrics.

Implementation Details. Our zero-shot framework requires no environment-specific training. For the high-level *Language Action* stage, we evaluate two leading MLLMs: Gemini-2.5-Pro and GPT-4o. For the *Vision Action* stage, we primarily use the efficient Qwen2.5-VL-32B, with other models explored in our ablation studies. The agent’s observation is composed of posed 640×480 RGB-D images, low-level path planning is executed using the Fast Marching Method (FMM) on a global map constructed from depth observations. All experiments were conducted on 8 NVIDIA RTX 4090 GPUs for parallel evaluation.

Inference Cost. To provide a clear picture of the computational cost, we report the token usage for GPT-4o + Qwen2.5-VL-32B over the 100-episode validation set. On average, each trajectory required approximately 32,682 tokens with 7.93 calls for the high-level planner (GPT-4o) and 8,050 tokens with 7.50 calls for the grounding model (Qwen2.5-VL-32B). Based on current API pricing, the total inference cost is approximately **\$0.084 USD per episode**. This highlights the efficiency of our hierarchical design, where an expensive, powerful model is used for high-impact decisions, while a lightweight model handles the low-level perceptual grounding task.

TABLE I: Main results on the VLN-CE benchmark. LaViRA significantly outperforms all previous zero-shot methods. **Best** and **second-best** zero-shot results are highlighted.

Method	NE↓	OSR↑	SR↑	SPL↑
Supervised Learning				
CMA [35]	6.92	45	37	32.2
RecBERT [35]	5.80	57	48	43.2
ETPNav [21]	5.15	58	52	52.2
BEVBert [36]	5.13	64	60	53.4
Zero-Shot				
Random	8.63	12	2	1.5
NavGPT-CE [37]	8.37	26.9	16.3	10.2
DiscussNav-CE [37]	7.77	15	11	10.5
MapGPT-CE [38]	8.16	21	7	5.0
Open-Nav [3]	6.70	23	19	16.1
SmartWay [4]	7.11	51	29	22.5
InstructNav [6]	6.89	47	31	24.0
CA-Nav [5]	7.58	48.0	25.3	10.8
GC-VLN [28]	7.30	41.8	33.6	16.3
LaViRA(GPT-4o)	6.43±0.28	43.3±3.2	36.0±1.7	28.3±0.8
LaViRA(Gemini-2.5-Pro)	6.54±0.27	48.7±2.1	38.3±0.6	28.3±0.9

B. Main Results

We compare LaViRA with existing methods on the VLN-CE benchmark. As shown in Table I, our method sets a new state-of-the-art for zero-shot VLN-CE.

The Gemini-2.5-Pro variant achieves SR of 38.3% and SPL of 28.3%, improving over the prior best zero-shot method InstructNav [6] by 7.3 points in SR and 4.3 points in SPL. Low standard deviations across runs highlight the framework’s robustness and stability, a key advantage for real-world applications. Notably, LaViRA surpasses supervised methods in SR, demonstrating the power of our hierarchical decomposition and MLLM reasoning. The GPT-4o variant also delivers strong, stable results. These findings validate our central hypothesis: decomposing the navigation task and leveraging advanced MLLM reasoning enables superior zero-shot performance without any pre-trained way-point predictor.

C. Ablation Studies

We performed a series of ablation studies to analyze LaViRA’s performance and quantify the contribution of its core components. Our baseline for these studies is LaViRA (GPT-4o + Qwen2.5-VL-32B). Although the Gemini-2.5-Pro variant delivered superior performance, we used the GPT-4o variant for ablations due to documented stability issues with the Gemini-2.5-Pro API during our experiments, which could have compromised the consistency of iterative testing. The GPT-4o model provided the necessary reliability for a rigorous and reproducible analysis.

Model Selection. Our framework is model-agnostic, allowing flexible combinations of MLLMs. As shown in Table II, our experiments confirm that pairing models of different scales according to task granularity is crucial. Using a powerful MLLM (e.g., GPT-4o) for high-level *Language Action* (LA) is key; replacing it with the smaller Qwen2.5-VL-72B leads to a 7.0-point drop in SPL. For the more focused *Vision Action* (VA) stage, an efficient model like Qwen2.5-VL-32B proves highly effective. This aligns with the principles of hierarchical VLN discussed in our related work II-C, where specialized modules handle different levels of the task.

TABLE II: Ablation on model selection. Performance is maximized when a powerful MLLM handles high-level Language Actions (LAM) and an efficient MLLM handles focused Vision Actions (VAM). Qwen-32B and Qwen-72B are short for Qwen2.5-VL-32B and Qwen2.5-VL-72B.

LAM	VAM	NE↓	OSR↑	SR↑	SPL↑
Qwen-32B	Qwen-32B	8.04±0.28	25.3±2.1	19.7±3.5	14.3±2.6
Qwen-72B	Qwen-32B	7.18±0.18	33.3±2.1	27.7±2.3	21.3±2.7
GPT-4o	Qwen-32B	6.43±0.28	43.3±3.2	36.0±1.7	28.3±0.8
GPT-4o	Qwen-72B	6.78±0.11	39.3±2.1	32.3±1.2	23.8±0.7
GPT-4o	GPT-4o	7.47±0.24	26.0±7.5	20.7±5.7	16.8±4.9

TABLE III: Ablation on framework design. A three-stage pipeline, rich visual history, and flexible backtracking are all crucial for robust navigation.

Configuration	NE↓	OSR↑	SR↑	SPL↑
LaViRA (Full)	6.43±0.28	43.3±3.2	36.0±1.7	28.3±0.8
Framework Decomposition				
w/o LA	8.94±0.53	13.0±3.0	6.7±0.6	4.4±1.2
w/o VA	7.28±0.23	34.0±4.0	23.0±5.2	13.9±3.4
w/o LA+VA	9.21±0.11	1.7±0.6	0.0±0.0	0.0±0.0
History Representation				
text obs. + act.	6.99±0.32	37.7±5.5	31.0±3.6	23.0±2.8
only act.	6.54±0.11	38.3±1.2	32.7±2.1	25.1±1.8
only visual obs.	6.64±0.23	36.0±5.6	29.3±3.5	22.5±2.2
only text obs.	6.96±0.24	35.7±4.0	27.7±2.5	21.8±1.8
w/o history	6.90±0.46	36.3±7.0	27.0±5.6	19.4±7.3
Backtracking Mechanism				
w/o backtrack	6.92±0.32	42.0±3.0	30.0±4.4	22.2±4.0
last waypoint only	6.65±0.16	41.7±6.0	31.7±1.5	23.5±1.0
Pixel Point Selection				
direct output point	7.00±0.84	45.3±3.1	31.0±5.6	19.4±5.3
bbox median depth	6.73±0.09	51.0±2.6	34.0±1.0	21.8±1.2

Interestingly, using a powerful model like GPT-4o for both stages significantly degrades performance (SPL drops from 28.3% to 16.8%). This suggests that simply using the largest model is not optimal, the best results—pairing a top-tier planning model with an efficient grounding model—validate our core hypothesis and demonstrate a key advantage of our zero-shot hierarchical approach.

Framework Design Choices. We conducted further ablations on key design choices, as shown in Table III. First, we ablated the **hierarchical structure**. An end-to-end baseline (“w/o LA+VA”) fails with 0% SPL. Removing the high-level planner (“w/o LA”) yields 4.4% SPL, while removing the perceptual grounding module (“w/o VA”) achieves 13.9% SPL. Our full framework outperforms the latter by 14.4 points, confirming the effectiveness of coarse-to-fine decomposition. Next, we analyzed **history representation**. Replacing visual history with textual descriptions drops SPL by 5.3 points to 23.0%. Using only past actions (25.1%) or only textual observations (21.8%) also underperforms, showing that combining raw visual observations and actions is most effective. We then evaluated the **backtracking mechanism**. Disabling it causes a 6.1-point SPL drop. Restricting backtracking to the last waypoint only remains 4.8 points worse than our flexible strategy. Finally, we ablated **pixel point selection**. Directly outputting pixel coordinates achieves 19.4% SPL, which is 8.9 points lower than our bounding-box + bottom-center heuristic approach.

D. Qualitative Analysis

To offer qualitative insights into LaViRA’s decision-making, Figure 4 shows a successful navigation run and common failures. In the success case, LaViRA demonstrates its coarse-to-fine approach: it first makes a high-level directional choice (“navigate left”), then grounds this in a visual landmark (“Black door with glass panels”), and finally selects a precise waypoint. This hierarchical process validates our design.

The failure cases illustrate three common errors: (1) A Language Action error from ambiguous instructions, e.g., failing to identify the correct door when multiple doors are visible. This can be mitigated by explicit sub-instruction decomposition during prompting. (2) A Vision Action error where the correct object is grounded to the wrong image region. Although VAM excels at bounding-box grounding, it sometimes fails on larger areas like “hallway” or “living room”. Integrating open-vocabulary segmentation (e.g., SAM 2) as post-processing can address this. (3) A simulation-induced depth error where transparent objects (e.g., windows in Habitat) lack depth values, leading to incorrect 3D projection. In real-world deployment, this can be resolved via LiDAR or stereo depth fusion.

V. REAL-WORLD EXPERIMENTS

To validate LaViRA’s practicality beyond simulation, we deployed it on two distinct real-world robots: a Unitree Go1 quadruped and an Agilix Cobot Magic wheeled platform. These experiments tested the framework’s sim-to-real transferability, requiring only the replacement of the low-level robot controller. The Unitree Go1 was equipped with a Jetson Orin NX and an Intel RealSense D435i camera, using its native velocity controller to navigate. The Agilix Cobot Magic platform used a chest-mounted Orbbec Dabai camera and a 2-DOF velocity controller for its mobile base. In both deployments, the onboard computers called the respective MLLM APIs for Language and Vision Actions, transmitting the resulting target pixel coordinates to the robot’s navigation system.

As shown in Figure 5, both platforms successfully executed navigation tasks in complex indoor office environments. The supplementary video further demonstrates consistent performance across multiple cases on each robot, highlighting the framework’s strong sim-to-real transferability with only platform-specific low-level controller replacement. These qualitative results confirm that LaViRA’s hierarchical reasoning generalizes effectively from simulation to physical hardware without any training.

VI. CONCLUSION

We introduce LaViRA, a hierarchical framework for zero-shot Vision-and-Language Navigation in continuous environments. By decomposing navigation into a coarse-to-fine action hierarchy, LaViRA eliminates pre-trained waypoint predictors and fully leverages multi-granularity MLLM reasoning. Experiments show that pairing a powerful MLLM for high-level planning with an efficient one for perceptual

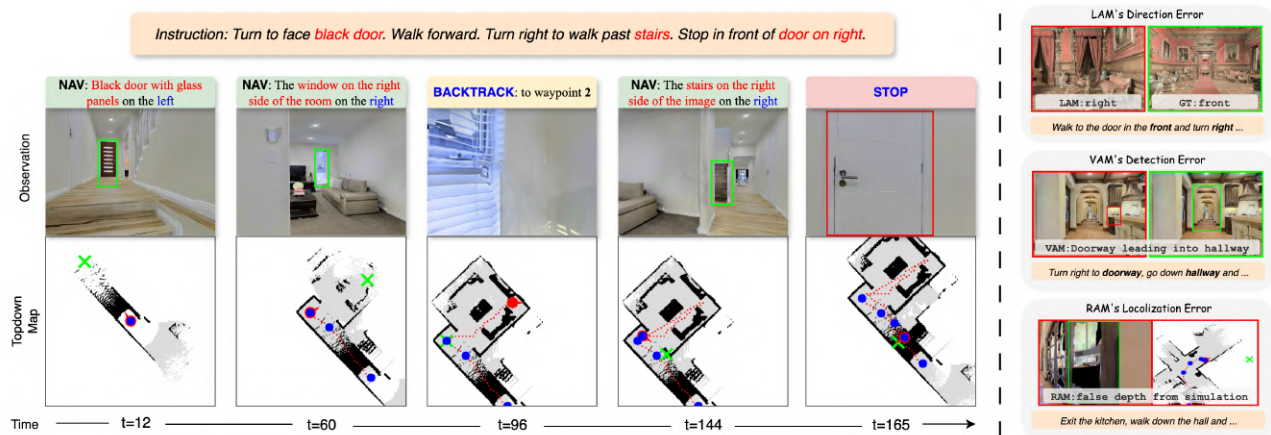


Fig. 4: **Visualization examples.** (Left) Navigation visualization: Language Action outputs are in blue text. Vision Action outputs bounding boxes in green and target descriptions in red. The robot’s pose is a red arrow, history waypoints are blue dots, and the next target position is marked by a green cross. (Right) Failure cases visualization: Language Action misjudges direction due to ambiguous instructions; Vision Action selects the wrong region despite correct target description; simulation reconstruction errors cause incorrect target localization.

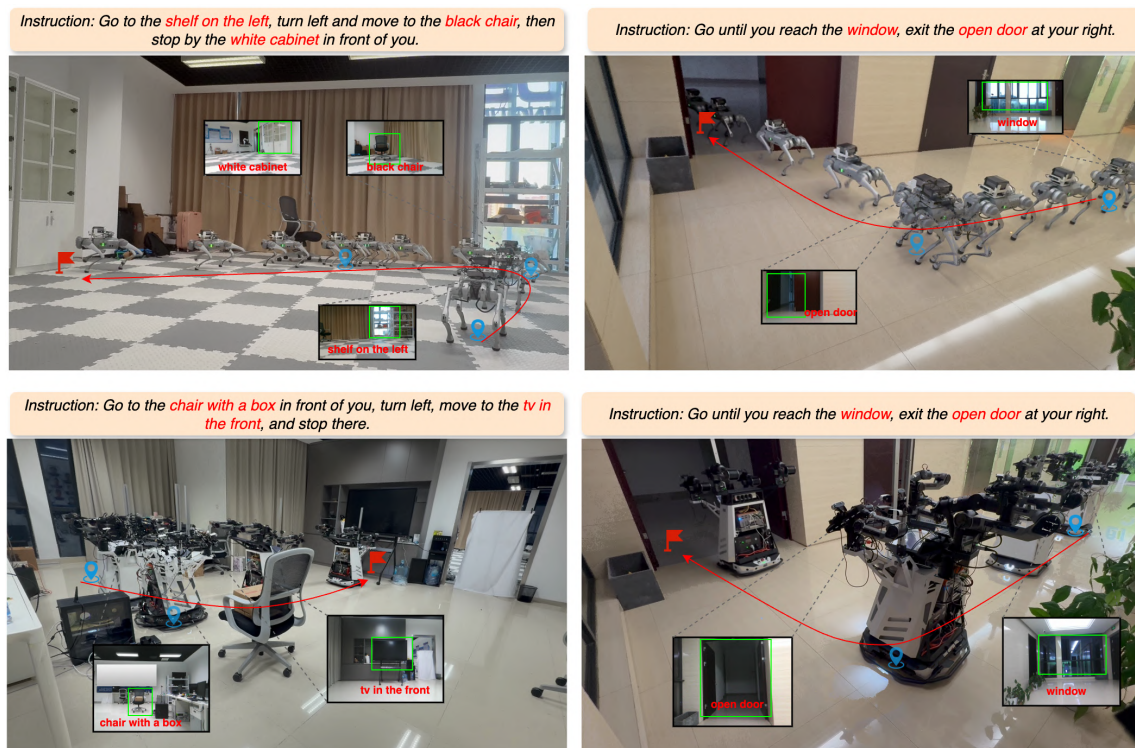


Fig. 5: **Real-world experiment examples.** LaViRA guides a Unitree Go1 quadruped (top) and an Agilex Cobot Magic wheeled robot (bottom) in an office. The visualization shows the third-person view of the robot’s trajectory alongside the agent’s ego view and targets, demonstrating successful real-world adaptation on diverse platforms.

grounding achieves state-of-the-art performance on the VLN-CE benchmark, significantly outperforming existing zero-shot methods.

Despite these results, LaViRA relies on proprietary MLLMs, raising concerns about API latency, cost volatility, stability, and limited control. Its performance ceiling is bounded by off-the-shelf models, as seen in failures on

ambiguous instructions and large-area grounding. Real-world deployment faces sensor noise, dynamic obstacles, varying illumination, and long-term drift. Future work will distill the pipeline into open-source MLLMs, add caching and adaptation mechanisms to reduce latency and cost, and integrate robust perception for reliable long-term deployment.

REFERENCES

- [1] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3674–3683.
- [2] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, "Beyond the nav-graph: Vision and language navigation in continuous environments," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 104–120.
- [3] Y. Qiao, W. Lyu, H. Wang, Z. Wang, Z. Li, Y. Zhang, M. Tan, and Q. Wu, "Open-nav: Exploring zero-shot vision-and-language navigation in continuous environment with open-source llms," *arXiv preprint arXiv:2409.18794*, 2024.
- [4] X. Shi, Z. Li, W. Lyu, J. Xia, F. Dayoub, Y. Qiao, and Q. Wu, "Smartway: Enhanced waypoint prediction and backtracking for zero-shot vision-and-language navigation," *arXiv preprint arXiv:2503.10069*, 2025.
- [5] K. Chen, D. An, Y. Huang, R. Xu, Y. Su, Y. Ling, I. Reid, and L. Wang, "Constraint-aware zero-shot vision-language navigation in continuous environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
- [6] Y. Long, W. Cai, H. Wang, G. Zhan, and H. Dong, "Instructnav: Zero-shot system for generic instruction navigation in unexplored environment," *arXiv preprint arXiv:2406.04882*, 2024.
- [7] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models," in *International Conference on Machine Learning*, 2023, pp. 19 730–19 742.
- [8] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.
- [9] R. S. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, no. 1–2, pp. 181–211, 1999.
- [10] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *arXiv preprint arXiv:1709.06158*, 2017.
- [11] Y. Qi, Q. Wu, P. Anderson, X. Wang, W. Y. Wang, C. Shen, and A. van den Hengel, "Reverie: Remote embodied visual referring expression in real indoor environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9979–9988.
- [12] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldrige, "Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding," *arXiv preprint arXiv:2010.07954*, 2020.
- [13] Y. Hong, Q. Wu, Y. Qi, C. Rodriguez-Opazo, and S. Gould, "Vln bert: A recurrent vision-and-language bert for navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1643–1653.
- [14] Y. Qi, Z. Pan, Y. Hong, M.-H. Yang, A. van den Hengel, and Q. Wu, "The road to know-where: An object-and-room informed sequential bert for indoor vision-language navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1635–1644.
- [15] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang, "Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6629–6638.
- [16] J. Li, H. Tan, and M. Bansal, "Envedit: Environment editing for vision-and-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 386–15 396.
- [17] D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L.-P. Morency, T. Berg-Kirkpatrick, K. Saenko, D. Klein, and T. Darrell, "Speaker-follower models for vision-and-language navigation," in *Advances in Neural Information Processing Systems*, 2018, pp. 3318–3329.
- [18] Z. Wang, J. Li, Y. Hong, Y. Wang, Q. Wu, M. Bansal, S. Gould, H. Tan, and Y. Qiao, "Scaling data generation in vision-and-language navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 12 009–12 020.
- [19] Y. Mei, Y. Yang, L. Guo, Q. Wang, M.-M. Yu, X. He, W. Wu, and J. Liu, "Urbannav: Learning language-guided urban navigation from web-scale human trajectories," *arXiv preprint arXiv:2512.09607*, 2025.
- [20] S. Chen, P.-L. Guhur, M. Tapaswi, C. Schmid, and I. Laptev, "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*, 2022, pp. 16 516–16 526.
- [21] D. An, H. Wang, W. Wang, Z. Wang, Y. Huang, K. He, and L. Wang, "Etpnav: Evolving topological planning for vision-language navigation in continuous environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–16, 2024.
- [22] Z. Wang, X. Li, J. Yang, Y. Liu, and S. Jiang, "Gridmm: Grid memory map for vision-and-language navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 15 625–15 636.
- [23] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
- [24] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.
- [25] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, 2021, pp. 8748–8763.
- [26] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soriccut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.
- [27] Y. Hong, Z. Wang, Q. Wu, and S. Gould, "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*, 2022, pp. 15 418–15 428.
- [28] H. Yin, H. Wei, X. Xu, W. Guo, J. Zhou, and J. Lu, "GC-VLN: Instruction as graph constraints for training-free vision-and-language navigation," *arXiv preprint arXiv:2509.10454*, 2025.
- [29] C. Gao, X. Peng, M. Yan, H. Wang, L. Yang, H. Ren, H. Li, and S. Liu, "Adaptive zone-aware hierarchical planner for vision-language navigation," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 14 911–14 920.
- [30] S. Zhang, X. Song, X. Yu, and *et al.*, "HOZ++: Versatile hierarchical object-to-zone graph for object navigation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
- [31] M. Z. Irshad, C.-Y. Ma, and Z. Kira, "Hierarchical cross-modal agent for robotics vision-and-language navigation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
- [32] F. Johnson, B. B. Cao, A. Ashok, and *et al.*, "Feudal networks for visual navigation," *arXiv preprint arXiv:2402.12498*, 2024.
- [33] L. Zhang, X. Hao, Y. Tang, and *et al.*, "NavA³: Understanding any instruction, navigating anywhere, finding anything," *arXiv preprint arXiv:2508.04598*, 2025.
- [34] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, *et al.*, "Habitat: A platform for embodied ai research," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9338–9346.
- [35] Y. Hong, Z. Wang, Q. Wu, and S. Gould, "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*, 2022, pp. 15 439–15 449.
- [36] D. An, Y. Qi, Y. Li, Y. Huang, L. Wang, T. Tan, and J. Shao, "Bevbert: Multimodal map pre-training for language-guided navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2737–2748.
- [37] Y. Long, X. Li, W. Cai, and H. Dong, "Discuss before moving: Visual language navigation via multi-expert discussions," in *IEEE International Conference on Robotics and Automation*, 2023, pp. 17 380–17 387.
- [38] J. Chen, B. Lin, R. Xu, Z. Chai, X. Liang, and K.-Y. K. Wong, "Mapgpt: Map-guided prompting with adaptive path planning for vision-and-language navigation," *arXiv preprint arXiv:2401.07314*, 2024.