# STRIDER: Navigation via Instruction-Aligned Structural Decision Space Optimization

**Diqi He**[1]*, **Xuehao Gao**[1]*, **Hao Li**[1,2], **Junwei Han**[1,3], **Dingwen Zhang**[1]†
[1]Northwestern Polytechnical University
[2]Nanyang Technological University
[3]Chongqing University of Posts and Telecommunications
https://github.com/diqihe666/STRIDER-Nav

## Abstract

The Zero-shot Vision-and-Language Navigation in Continuous Environments (VLN-CE) task requires agents to navigate previously unseen 3D environments using natural language instructions, without any scene-specific training. A critical challenge in this setting lies in ensuring agents' actions align with both spatial structure and task intent over long-horizon execution. Existing methods often fail to achieve robust navigation due to a lack of structured decision-making and insufficient integration of feedback from previous actions. To address these challenges, we propose STRIDER (Instruction-Aligned Structural Decision Space Optimization), a novel framework that systematically optimizes the agent's decision space by integrating spatial layout priors and dynamic task feedback. Our approach introduces two key innovations: 1) a Structured Waypoint Generator that constrains the action space through spatial structure, and 2) a Task-Alignment Regulator that adjusts behavior based on task progress, ensuring semantic alignment throughout navigation. Extensive experiments on the R2R-CE and RxR-CE benchmarks demonstrate that STRIDER significantly outperforms strong SOTA across key metrics; in particular, it improves Success Rate (SR) from 29% to 35%, a relative gain of 20.7%. Such results highlight the importance of spatially constrained decision-making and feedback-guided execution in improving navigation fidelity for zero-shot VLN-CE.

## 1 Introduction

VLN-CE challenges agents to follow natural language instructions to navigate previously unseen 3D environments without any scene-specific training or fine-tuning [42, 25, 5, 60, 35]. This task is a critical benchmark for embodied AI, requiring agents to generalize perception, reasoning, and action across diverse and dynamic scenes [3, 24, 40, 62, 47, 8]. In comparison to discrete VLN tasks [3, 51, 26], VLN-CE more closely reflects real-world deployment conditions, where agents must process RGB-D inputs and make continuous movement decisions [18, 61, 21]. As a result, zero-shot VLN-CE pushes the boundaries of language-grounded generalization in embodied navigation [38, 61, 35, 5].

A central challenge in VLN-CE lies not only in grounding instructions into perception, but also in ensuring that the agent's behavior remains aligned with the semantic intent of the instruction throughout the navigation process. In unfamiliar environments, agents may correctly understand the instruction yet still exhibit execution drift [47, 34, 42, 8, 18, 35], such as stopping near the target

*Equal contribution.
†Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

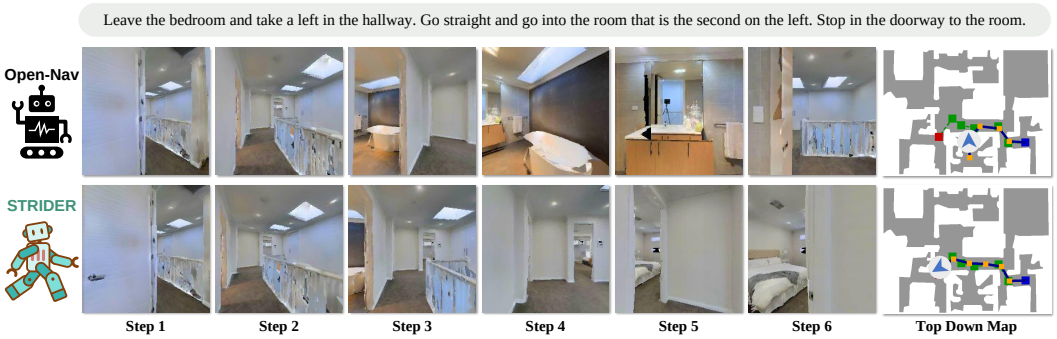


Figure 1: **Navigation behavior comparison between STRIDER and Open-Nav [42].** Given the same instruction, Open-Nav demonstrates execution drift, such as prematurely turning away from a hallway, and accumulates deviations over time. In contrast, STRIDER generates trajectories that more accurately follow the intended path and reach the goal region.

room without entering or prematurely turning away from a hallway, as shown in Fig. 1. These failures highlight a significant gap between *what the agent understands* and *how it acts*. Our key insight is that effective zero-shot VLN-CE agents must go beyond strong perception and reasoning——they must operate within an *Instruction-Aligned Structural Decision Space*, a decision space that is explicitly structured by the environment and continuously regulated based on task progress.

However, existing approaches typically rely on learned waypoint predictors or sequence-to-sequence policies that map instructions and visual inputs directly to actions [42, 24, 18, 8]. While effective at modeling local navigability, these models tend to ignore structured representations that capture the global layout or semantic task progression [15, 16, 41, 26, 34]. Moreover, these methods often operate in an open-loop fashion, inferring each action independently without feedback on prior decisions [2, 9, 56, 38, 61]. This limitation hinders their ability to assess whether actions have brought the agent closer to the goal, leading to deviations from the instruction's intent, especially in complex or ambiguous scenes. While instruction grounding has seen significant progress, insufficient attention has been paid to optimizing the agent's decision space in a manner that aligns both with spatial structure and task instructions.

To address these challenges, we introduce **STRIDER**, a zero-shot VLN-CE framework built on the principle of *Instruction-Aligned Structural Decision Space Optimization*. Our approach is grounded in the observation that semantic misalignment often arises not from perceptual misunderstanding, but from the inability to maintain alignment with both the spatial structure of the environment and the task's semantic progress over long-horizon execution [3, 23]. STRIDER adopts a modeling-first approach: instead of directly predicting actions from visual inputs and instructions, it focuses on optimizing the agent's decision space by integrating spatial structure and task progress awareness [50]. By embedding spatial layout priors and continuous goal feedback into the agent's decision-making process, STRIDER enables the agent to navigate within paths that are both spatially coherent and semantically aligned with the task. This dynamic adjustment of behavior reduces execution drift and improves instruction fidelity, enhancing both spatial generalization and instruction-level adherence.

STRIDER achieves this optimization through two tightly integrated modules. The **Structured Waypoint Generator** creates a layout-constrained action space by extracting skeletons from depth-based navigable regions [10, 36, 45]. This module ensures that the agent's movement decisions are limited to paths that are both spatially coherent and meaningful, grounded in the environment's structure. The **Task-Alignment Regulator** continuously monitors task progress and adjusts the agent's behavior accordingly, ensuring actions remain aligned with the instruction-defined goal. It recalibrates behavior whenever deviations are detected, while staying within the spatial constraints defined by the structured action space. Together, these modules optimize the agent's decision space by structuring it with spatial constraints and regulating it according to task progress, ultimately enabling more efficient and semantically faithful navigation.

Our contributions are summarized as follows:

- We propose **STRIDER**, a zero-shot VLN-CE framework based on the principle of *Instruction-Aligned Decision Space Optimization*. STRIDER optimizes the agent's decision space by integrating spatial structure and task feedback, enabling more coherent and instruction-faithful navigation behavior.
- STRIDER consists of two tightly coupled modules that jointly optimize the agent's decision space: (1) A **Structured Waypoint Generator** that constructs a layout-constrained planning space from depth-based skeletons, embedding spatial priors into the action space; (2) A **Task-Alignment Regulator** that monitors semantic progress across steps and adjusts behavior accordingly, ensuring actions remain aligned with instruction goals.
- We evaluate STRIDER on two standard zero-shot VLN-CE benchmarks, R2R-CE and RxR-CE, where it consistently outperforms strong baselines on core navigation metrics, demonstrating the benefit of structuring and regulating the agent's decision space.

## 2 Related Work

### 2.1 Vision-and-Language Navigation

Significant progress has been made in Vision-and-Language Navigation (VLN), especially in continuous settings [42, 24, 25, 18, 39, 3]. VLN approaches can be broadly categorized into supervised learning with environment-specific training and zero-shot generalization. Supervised methods, such as imitation learning and reinforcement learning [53], rely on human-annotated trajectories to train navigation policies [2, 39, 13, 58, 59, 19], often enhancing performance through visual-language alignment, attention modules, auxiliary tasks [39, 50, 62, 1], and explicit or implicit 3D scene understanding and reconstruction [55, 31, 29, 30, 14, 32]. On the other hand, zero-shot methods aim to generalize to unseen scenes without task-specific fine-tuning. These approaches typically leverage instruction tuning, pretrained Vision-Language Models (VLMs) or Large Language Models (LLMs) [42, 37, 46, 54, 48]. While zero-shot methods show promise, they primarily focus on local observations and immediate actions, lacking global spatial awareness and long-term planning capabilities. Few methods integrate feedback mechanisms to monitor task progress or correct deviations, limiting their adaptability in complex scenarios.

### 2.2 Decision Space Optimization

Decision Space Optimization originates from robotics and autonomous systems, where it involves structuring an agent's decision-making within a "decision space" that integrates spatial constraints, task goals, and feedback from the environment [27, 52, 22, 33]. This structured decision-making allows agents to balance short-term actions with long-term objectives, improving task execution. In robotics, decision space optimization ensures that actions are spatially coherent and consistent with global goals, aiding navigation and manipulation in dynamic environments [49, 43, 11]. In the field of VLN, early methods primarily map instructions to actions based on local observations but struggle with long-term planning and generalization to unseen environments [3, 13, 39]. More recent methods have incorporated memory, attention mechanisms, and reinforcement learning to improve decision-making [53, 17, 41]. These approaches also use visual-language alignment for action prediction and introduce curriculum learning for enhanced performance [62, 2, 50]. However, challenges remain in maintaining global spatial awareness and ensuring long-term task alignment. STRIDER addresses these issues through Instruction-Aligned Structural Decision Space Optimization. It optimizes decision-making by ensuring actions are consistent with both spatial constraints and task goals, enabling better navigation performance over long horizons.

## 3 Method

In this section, we explore how VLN-CE can be enhanced by structuring the decision space through spatial constraints and integrating task feedback. We begin with an overview of the VLN-CE task and introduce our proposed STRIDER framework (Sec. 3.1). We then present two core components of our approach: (1) the use of local perception to construct spatially organized representations that capture region connectivity (Sec.3.2), and (2) the integration of task feedback to monitor subgoal progress and regulate action selection during execution (Sec. 3.3).

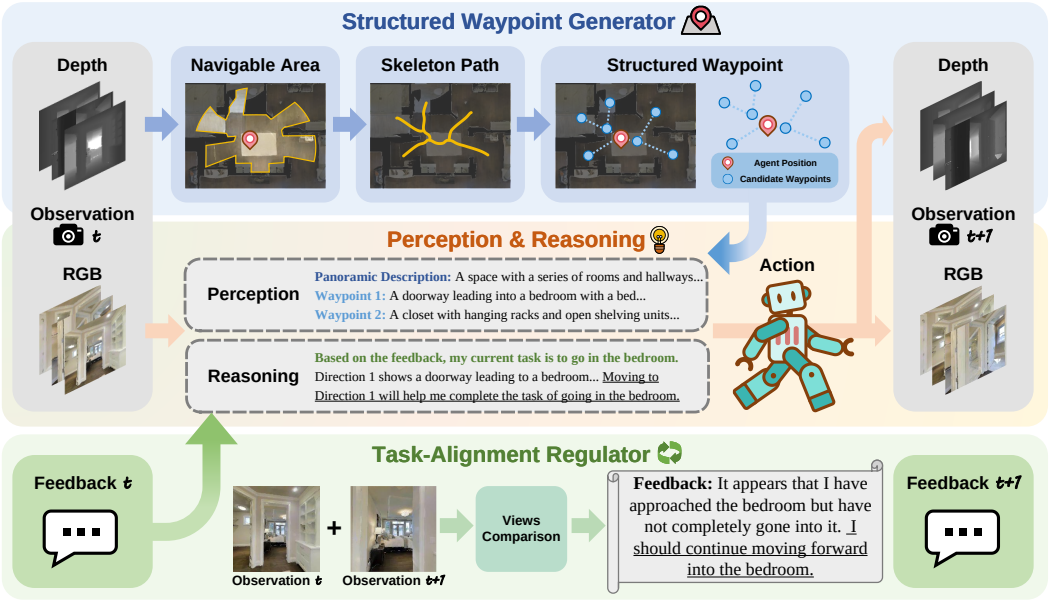


Figure 2: **Overview of the STRIDER pipeline.** The Structured Waypoint Generator constructs a layout-constrained waypoint space by extracting skeleton paths from navigable depth observations. The agent performs perception and reasoning over visual descriptions and feedback to identify suitable actions in context. To maintain semantic alignment over time, the Task-Alignment Regulator compares current and previous observations and generates feedback that guides the next action.

### 3.1 Overview

**Task Definition.** In the task of zero-shot VLN-CE, the agent begins at a specified starting location and must reach a target destination by interpreting both verbal guidance and visual observations. At each timestep, it receives a panoramic 360° view composed of 12 RGB-D images captured at fixed intervals (0°, 30°, ..., 330°). From these observations, the agent predicts a set of waypoints, candidate navigable locations, and selects one to move toward. The episode continues until the instruction is fulfilled or the agent reaches the goal.

**STRIDER.** Our proposed STRIDER framework follows a zero-shot VLN-CE pipeline, as illustrated in Fig. 2. At each timestep $t$, the agent receives an RGB-D observation $\mathbf{O}_t = (\mathbf{I}_t, \mathbf{D}_t)$ and a language instruction $L$. The Structured Waypoint Generator processes the depth $\mathbf{D}_t$ to extract navigable regions and generates a set of structured waypoints $\mathcal{W}_t$ organized by spatial connectivity. Subsequently, a pretrained Vision-Language Model (VLM) describes the RGB input $\mathbf{I}_t$, attending to visual content in the directions of candidate waypoints $\mathcal{W}_t$ to form a decision space $\mathcal{A}_t$. A Large Language Model (LLM) then reasons over the instruction $L$, the current decision space $\mathcal{A}_t$, and the task feedback $f_t$ generated from the previous step to select a waypoint $w_t^* \in \mathcal{W}_t$ for movement.

After executing the action toward $w_t^*$, the agent receives the next observation $\mathbf{O}_{t+1}$. The Task-Alignment Regulator compares $\mathbf{O}_t$ and $\mathbf{O}_{t+1}$ using the VLM to detect progress toward the instruction goal and generates updated feedback $f_{t+1}$. This feedback is leveraged in the next decision step, completing a closed-loop control cycle grounded in structured perception and task alignment.

### 3.2 Structured Waypoint Generator

To optimize the agent's decision space in continuous environments, we generate a layout-constrained set of candidate actions that explicitly reflect the spatial structure of the scene. Rather than relying on local navigability or unconstrained policy outputs, we propose a **Structured Waypoint Generator** that transforms depth input into a compact topological graph of navigable options. This process consists of three stages: (1) navigable region extraction, (2) topological skeleton abstraction, and (3) structured waypoint selection.

**Navigable Region Extraction.** At time step $t$, the agent receives a panoramic RGB-D observation $\mathbf{O}_t = (\mathbf{I}_t, \mathbf{D}_t)$, where $\mathbf{D}_t$ denotes multi-view depth input. Then we reconstruct a local point cloud $\mathcal{P}_t \subset \mathbb{R}^3$, where each point $\mathbf{p}_i = (x_i, y_i, z_i)$ represents a 3D coordinate in the agent-centric reference frame. To isolate feasible movement areas, we filter for ground-level points and project them into a 2D top-down plane:

$$\Omega_t = \Pi\left(\{\mathbf{p}_i \in \mathcal{P}_t \mid \mathbf{p}_i(z) < \delta_h,\ \|\mathbf{p}_i(x,y)\| < r\}\right), \tag{1}$$

where $\delta_h$ is the height threshold, $r$ is the local planning radius, and $\Pi(\cdot)$ denotes orthographic projection onto the horizontal plane. $\Omega_t$ defines the local traversable area around the agent.

**Topological Skeleton Extraction.** To introduce structural priors into the local decision space, we abstract the raw traversable region $\Omega_t$ into a topological skeleton $\mathcal{S}_t$ using morphological thinning [28]:

$$\mathcal{S}_t = Skeletonize(\Omega_t), \tag{2}$$

where $\mathcal{S}_t \subset \Omega_t$ approximates the center axis of free space. Rather than interpreting navigable space as a dense, unstructured area, the skeleton captures its underlying topology by tracing the central lines of movement through open regions. This is analogous to how humans often form mental maps of environments based on key corridors, intersections, and doorways, rather than memorizing complete spatial coverage. By reasoning over this abstract structure, the agent can better plan paths that respect the environment's layout and reduce unnecessary action noise.

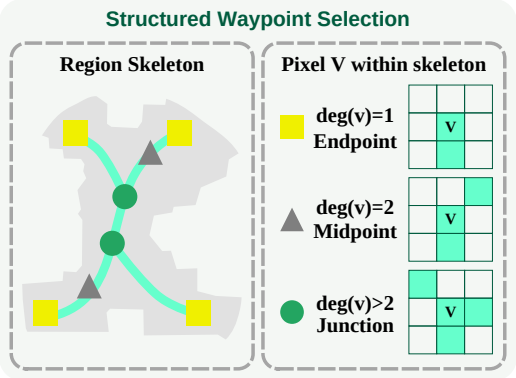


Figure 3: **Structured waypoint selection based on skeleton.** We categorize skeleton nodes by their degree and select only degree-1 endpoints as candidate waypoints.

**Structured Waypoint Selection.** We model the skeleton $\mathcal{S}_t$ as an undirected graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$, where nodes $\mathcal{V}_t$ correspond to skeleton pixels and edges $\mathcal{E}_t$ reflect 8-connected adjacency (including horizontal, vertical, and diagonal neighbors in the 2D grid). To reduce redundancy and focus on directionally meaningful locations, we select a sparse set of endpoints from the graph as candidate waypoints. Specifically, we retain only nodes with degree 1:

$$\mathcal{W}_t = \{v_i \in \mathcal{V}_t \mid \deg(v_i) = 1\}, \tag{3}$$

which correspond to the outermost reachable points on the local skeleton, as shown in Fig. 3. These endpoints naturally capture the agent's forward navigability and latent path divergence. Although we do not explicitly select junctions, their future branches are represented by the endpoints along each subpath.

Each selected waypoint $w_i \in \mathcal{W}_t$ is projected back into 3D. We input the RGB views $\mathbf{I}_t$ into a VLM to extract semantic information, focusing on visual content in the direction of each waypoint. For each $w_i$, the VLM outputs a textual description $\mathcal{D}_i$ of the corresponding direction. The final decision space is defined as a set of paired spatial-semantic candidates:

$$\mathcal{A}_t = \{(w_i, \mathcal{D}_i) \mid w_i \in \mathcal{W}_t,\ \mathcal{D}_i = \text{VLM}(\mathbf{I}_t, w_i)\}, \tag{4}$$

which couples structural layout with perceptual grounding. This layout-constrained action space forms part of the input for the LLM.

### 3.3 Task-Aligned Feedback Regulation

To ensure instruction-aligned behavior over long-horizon trajectories, we introduce a feedback regulation mechanism that dynamically adjusts the agent's decision space based on recent observations and subtask progression. This module plays a central role in our principle of *Instruction-Aligned Decision Space Optimization*, allowing the agent to refine its actions not only based on spatial layout but also on semantic alignment with the instruction over time.

**Visual Feedback Generation.** Through LLM reasoning, the agent selects a waypoint $w_t^*$ from the decision space $\mathcal{A}_t$, resulting in an action $a_t$. After executing $a_t$, it receives the next observation

$\mathbf{O}_{t+1} = (\mathbf{I}_{t+1}, \mathbf{D}_{t+1})$. To assess whether the agent has made progress toward the current subtask $\mathcal{T}_t$ (derived from instruction $L$), we input the observation pair $(\mathbf{O}_t, \mathbf{O}_{t+1})$ and the subtask $\mathcal{T}_t$ into the VLM to generate a feedback signal:

$$f_{t+1} = \text{VLM}(\mathbf{O}_t, \mathbf{O}_{t+1}, \mathcal{T}_t). \tag{5}$$

where $f_{t+1}$ is a textual reflection describing the change in scene with respect to $\mathcal{T}_t$—e.g., "partially entered the bedroom" or "moved away from the target." The feedback offers fine-grained progress estimation at each step, allowing the agent to detect incremental advances or errors.

**Feedback-Guided Action Selection.** To determine the next action, the agent reasons over the updated structured decision space $\mathcal{A}_{t+1}$ (similarly defined in Eq. (4)), the full instruction $L$, and the generated feedback $f_{t+1}$. The LLM outputs the next action via:

$$w^*_{t+1} = \text{LLM}(\mathcal{A}_{t+1}, f_{t+1}, L), \tag{6}$$

$$a_{t+1} = Action(w^*_{t+1}). \tag{7}$$

This loop forms a closed decision-feedback cycle, in which action selection is continuously guided by semantic progress monitoring. As the agent moves through the environment, each observation is not only encoded visually but interpreted in light of task intent, enabling corrective adjustments and reducing cumulative drift. In this way, the decision space is adaptively regulated by execution context, maintaining semantic coherence with the instruction over time. By iterating this closed-loop process for $T$ steps, the agent produces an action sequence $\{a_1, a_2, \ldots, a_T\}$ that successfully navigates toward the instruction-aligned target.

# 4 Experiments

## 4.1 Experimental Setup

**R2R-CE Dataset.** We conduct experiments on the R2R-CE dataset, which extends the Room-to-Room (R2R) benchmark for visual language navigation (VLN) [3, 25]. This dataset consists of natural language instructions paired with navigation trajectories in realistic 3D indoor environments, derived from the Matterport3D dataset [4]. We follow the settings of OpenNav [42], conducting tests on 100 randomly selected episodes from the dataset. In these experiments, we leverage both VLM and LLM to perform zero-shot navigation. Our goal is to fully leverage the generalization and reasoning capabilities of these pretrained models, enabling them to adapt to the current task without any additional training.

**RxR-CE Dataset.** We also use the RxR-CE dataset, which extends the Room-Across-Room (RxR) benchmark with similar challenging conditions [26, 25]. RxR features longer and more diverse instructions across multiple languages and emphasizes global navigation capabilities. The CE variant (Continuously Evolving) simulates viewpoint changes and environmental variations, making it suitable for evaluating the robustness and generalization of navigation agents under distribution shifts.

**Evaluation metrics.** We evaluate navigation performance using standard metrics. Navigation Error (NE) measures the shortest-path distance between the agent's final position and the goal. Success Rate (SR) is the percentage of episodes where the agent stops within 3 meters of the goal. Success weighted by Path Length (SPL) balances success and path efficiency [3]. Normalized Dynamic Time Warping (NDTW) reflects the similarity between the predicted and reference trajectories [20]. Oracle Success Rate (OSR) indicates the best possible success assuming the agent stops optimally along its path [25]. Trajectory Length (TL) records the average length of agent trajectories. Soft-DTW (SDTW) is a relaxed version of DTW that tolerates slight deviations in trajectory matching [12].

**Implementation details.** All experiments are conducted in simulated VLN-CE environments. At each step, the agent receives an RGB-D observation, where the RGB input is resized to $244 \times 244 \times 3$ and the depth map to $256 \times 256$. Structured waypoints are generated by extracting skeletons from depth without relying on any pretrained waypoint predictor. For perception and feedback generation, we use Qwen-VL-Max as the Vision-Language Model (VLM). The action selection process is guided by GPT-4o, which reasons over the instruction, structured perception, and feedback to choose the next waypoint. Our VLM and LLM are accessed via API rather than deployed locally; for local deployment using open-source models, please refer to Open-Nav [42].

Table 1: **Performance comparison on the R2R-CE dataset.** The dash "–" indicates that the corresponding metric was not reported in the original work. **Bold** indicates the best result. We report the relative percentage change of our method compared to the previous SOTA in parentheses. Red denotes improvement, while Green indicates degradation.

| Method | NE↓ | NDTW↑ | OSR↑ | SR↑ | SPL↑ |
|---|---|---|---|---|---|
| **Supervised Learning** | | | | | |
| CMA [25] | 6.92 | 50.77 | 45 | 37 | 32.17 |
| RecBERT [57] | 5.8 | 54.81 | 57 | 48 | 43.22 |
| BEVBert [1] | 5.13 | 61.40 | 64 | 60 | 53.41 |
| ETPNav [2] | 5.15 | 61.15 | 58 | 52 | 52.18 |
| HNR [55] | 4.42 | - | 67 | 61 | 51 |
| **Zero-Shot** | | | | | |
| Random [42] | 8.63 | 34.08 | 12 | 2 | 1.50 |
| LXMERT [25] | 10.48 | 18.73 | 22 | 2 | 1.87 |
| DiscussNav-GPT4 [38] | 7.77 | 42.87 | 15 | 11 | 10.51 |
| Open-Nav-Llama3.1 [42] | 7.25 | 44.99 | 23 | 16 | 12.90 |
| Open-Nav-GPT4 [42] | **6.70** | 45.79 | 23 | 19 | 16.10 |
| SmartWay [46] | 7.01 | - | **51** | 29 | 22.46 |
| **Ours** | 6.91(3.1%) | **51.8**(13.2%) | 39(23.5%) | **35**(20.6%) | **30.30**(34.9%) |

Table 2: **Performance comparison on the RXR-CE dataset.** The dash "-" indicates that the corresponding metric was not reported in the original work. **Bold** indicates the best result. We report the relative percentage change of our method compared to the previous SOTA in parentheses. Red denotes improvement, while Green indicates degradation.

| Method | NE↓ | NDTW↑ | SDTW↑ | SR↑ | SPL↑ |
|---|---|---|---|---|---|
| **Supervised Learning** | | | | | |
| LAW [44] | 11.04 | 37.0 | 8.0 | 10.0 | 9.0 |
| VLN↻BERT [19] | 8.98 | 46.7 | - | 27.1 | 23.7 |
| GridMM [56] | 8.42 | 48.2 | 33.7 | 36.3 | 30.1 |
| ETPNav [2] | 5.64 | 61.9 | 45.3 | 54.8 | 44.9 |
| WS-MGMap [6] | 9.83 | - | - | 15.0 | 12.1 |
| HNR [55] | 5.51 | 63.56 | 47.24 | 56.39 | 46.73 |
| **Zero-Shot** | | | | | |
| $A^2$Nav [7] | - | - | - | 16.8 | 6.3 |
| CA-Nav [5] | **10.37** | 13.5 | 5.0 | 19.0 | 6.0 |
| **Ours** | 11.19(7.9%) | **30.1**(122.9%) | **8.9**(78.0%) | **21.2**(11.5%) | **9.6**(52.3%) |

### 4.2 Main Results on Zero-Shot VLN-CE

We evaluate STRIDER on two standard zero-shot VLN-CE benchmarks: R2R-CE (Tab. 1) and RxR-CE (Tab. 2). Across both datasets, STRIDER consistently outperforms prior zero-shot methods on key metrics such as SPL, SR, and NDTW, indicating more reliable goal completion and higher trajectory fidelity. While supervised methods benefit from task-specific training, STRIDER remains competitive despite operating without fine-tuning or environment-specific adaptation.

On R2R-CE, STRIDER achieves substantial improvements in SPL and NDTW over all prior zero-shot models, driven by two key design factors. The Structured Waypoint Generator constrains the agent's behavior to layout-consistent paths, reducing detours and spatial drift, while the Task-Alignment Regulator provides real-time feedback to maintain semantic alignment and correct deviations. This combination enables STRIDER to balance spatial feasibility and instruction adherence, leading to gains across both path-quality and goal-completion metrics. On the more diverse RxR-CE benchmark, STRIDER continues to outperform other zero-shot methods, though with narrower margins—likely due to RxR's higher linguistic and geographic variability. Even in such settings, STRIDER's structured decision space offers a strong prior that compensates for ambiguity, while the feedback

Table 3: **Effect of Structured Waypoint Generator (SWG) under different node degree configurations. Bold** indicates the best result. We report the relative percentage change compared to the baseline in parentheses. Red denotes improvement, while Green indicates degradation. The gray-shaded row denotes the primary experimental configuration used in our main results.

| SWG | Node Degree | NE↓ | NDTW↑ | OSR↑ | SR↑ | SPL↑ |
|---|---|---|---|---|---|---|
| ✗ | ✗ | 7.19 | 48.78 | 29 | 24 | 21.07 |
| ✓ | 1 | 6.91(3.9%) | **51.87**(6.3%) | **39**(34.5%) | **35**(45.8%) | **30.30**(43.8%) |
| ✓ | $>2$ | 7.34(2.1%) | 49.88(2.3%) | 33(13.8%) | 28(16.7%) | 25.02(18.7%) |
| ✓ | $\neq 2$ | **6.83**(5.0%) | 51.12(4.8%) | 38(31.0%) | 33(37.5%) | 29.21(38.6%) |

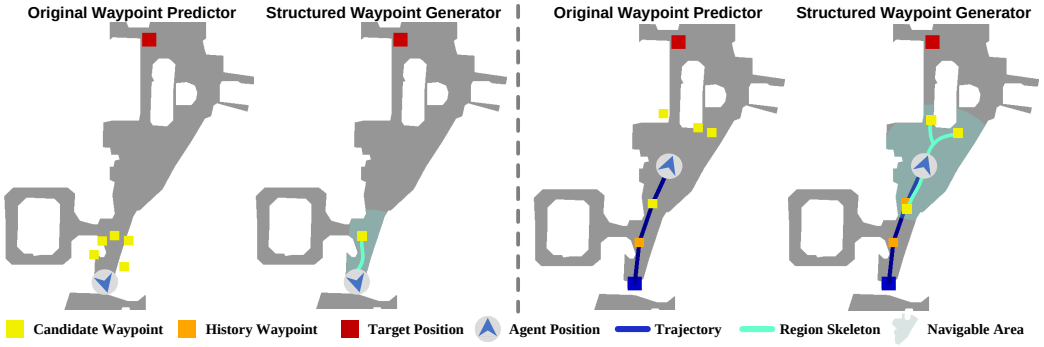


Figure 4: **Comparison between original waypoint predictor and Structured Waypoint Generator.** Our Structured Waypoint Generator extracts layout-consistent waypoints that align with the environment's topology, resulting in trajectories that are more goal-directed and spatially coherent.

loop supports adaptive planning. Together, these elements demonstrate the robustness and generality of decision space optimization under zero-shot conditions.

It is worth mentioning that STRIDER does not achieve the lowest Navigation Error (NE), which is expected given its emphasis on instruction alignment and structural feasibility over exact endpoint proximity. The agent may stop at semantically appropriate locations slightly offset from the goal, reflecting a preference for coherent, interpretable paths over shortcut-based precision—an acceptable tradeoff in instruction-guided navigation tasks.

### 4.3 Ablation Study

We conduct ablation experiments to analyze the impact of STRIDER's two core modules: the Structured Waypoint Generator and the Task-Alignment Regulator. For each module, we present both quantitative comparisons and qualitative visualizations to highlight their effects on navigation performance.

#### 4.3.1 Effect of Structured Waypoint Generator

We evaluate different configurations of the Structured Waypoint Generator to understand how layout-aware guidance affects agent behavior. As shown in Tab. 3, we consider two key design choices: (1) whether to use the Structured Waypoint Generator, and (2) how to guide waypoint selection based on skeleton connectivity—e.g., using only endpoint nodes ($\deg(v_i)$=1), higher-degree junctions ($\deg(v_i)$>2) or both. We use the original waypoint predictor trained on the R2R dataset as the baseline for STRIDER without the Structured Waypoint Generator. We present the visualization comparison between the original waypoint predictor and the Structured Waypoint Generator in Fig. 4.

Using degree-1 nodes (endpoints) yields the best overall performance across all metrics, as these points typically lie at corridor tips or face the goal direction, providing clear guidance with minimal ambiguity. In contrast, using only high-degree nodes ($>2$), such as junctions, leads to degraded performance with higher NE (7.34) and lower SR/SPL, likely due to increased behavioral uncertainty.

Table 4: **Effect of the Task-Alignment Regulator (TAR) on navigation performance. Bold** indicates the best result. We report the relative percentage change compared to the baseline in parentheses. Red denotes improvement, while Green indicates degradation. The gray-shaded row denotes the primary experimental configuration used in our main results.

| TAR | NE↓ | NDTW↑ | OSR↑ | SR↑ | SPL↑ |
|---|---|---|---|---|---|
| ✗ | **6.77** | 51.06 | **42** | 29 | 26.11 |
| ✓ | 6.91(2.1%) | **51.87**(1.6%) | 39(7.1%) | **35**(20.7%) | **30.30**(16.0%) |

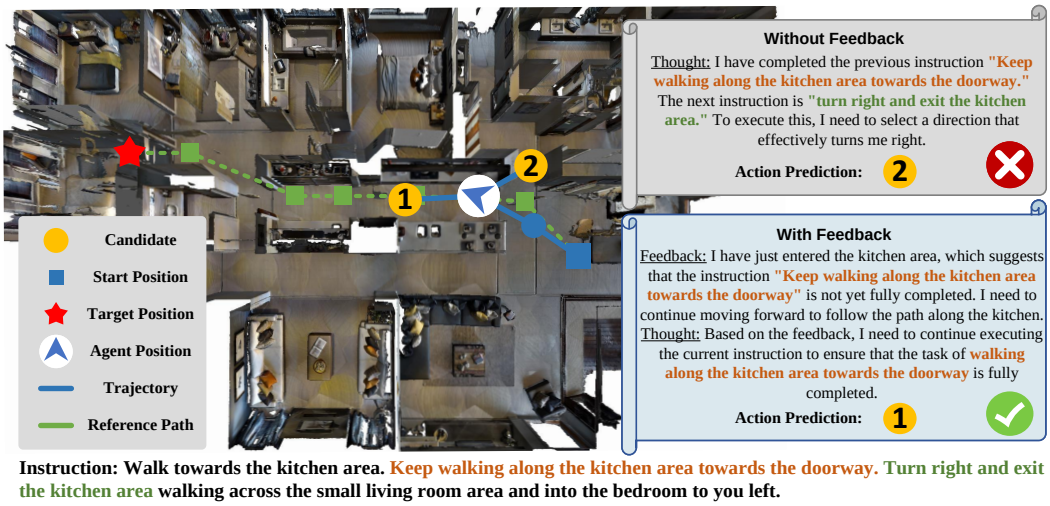


Figure 5: **Comparison of agent behavior under no-feedback and feedback-driven execution strategies.** Without feedback, the agent prematurely infers task completion, resulting in an incorrect action (Action 2). With feedback, the agent leverages the intermediate state to refine its understanding, yielding a more semantically consistent action (Action 1) aligned with the instruction.

Combining non-2-degree nodes (i.e., $\neq 2$) provides a good compromise, incorporating both endpoints and informative junctions to achieve strong performance (NE: 6.83, SPL: 29.21) at the cost of a slightly longer trajectory. Based on these results, we adopt the degree-1 configuration as the default in our main experiments.

#### 4.3.2 Effect of Task-Alignment Regulator

As shown in Tab. 4, enabling the Task-Alignment Regulator (TAR) leads to consistent improvements across SPL, SR, and NDTW, indicating that feedback-driven behavior adjustment helps the agent maintain semantic alignment and recover from drift during long-horizon navigation. While NE slightly increases (6.77→6.91), this reflects a more conservative execution pattern, where the agent prioritizes instruction fidelity and avoids premature termination, ultimately leading to higher task success and trajectory consistency. We illustrate the impact of the Task-Alignment Regulator on navigation behavior in Fig. 5. Overall, TAR enhances instruction fidelity without compromising spatial plausibility and is adopted in our main model configuration.

#### 4.3.3 Effect of Model-Agnostic Design

STRIDER is not tied to any single model and can operate effectively across components of similar capabilities. To verify this, we conduct experiments using various VLMs of different sizes and providers. As shown in Tab. 5, GPT-4o achieves the best performance in terms of NE (6.75) and SR (36). However, STRIDER also shows competitive results using similar models, such as Qwen-VL-Max and Qwen2.5-VL-72B, which perform strongly in NDTW, OSR and SR, while still maintaining an overall solid performance. Additionally, STRIDER continues to deliver good results with smaller models like Qwen2.5-VL-32B and Qwen2.5-VL-7B. The consistent performance across different

Table 5: **Ablation on different VLMs.** We test our method using various VLMs of different sizes and capabilities. **Bold** indicates the best result. The gray-shaded row denotes the primary experimental configuration used in our main results.

| VLM | TL | NE↓ | NDTW↑ | OSR↑ | SR↑ | SPL↑ |
|---|---|---|---|---|---|---|
| Qwen-VL-Max | 8.13 | 6.91 | 51.87 | **39** | 35 | 30.30 |
| Qwen2.5-VL-72B | 8.30 | 6.78 | 51.99 | 39 | 34 | 29.07 |
| Qwen2.5-VL-32B | 8.56 | 7.12 | 48.02 | 33 | 28 | 24.20 |
| Qwen2.5-VL-7B | 8.92 | 7.46 | 46.35 | 29 | 24 | 21.12 |
| GPT-4o | 8.01 | **6.75** | 50.12 | 39 | **36** | **31.37** |
| Gemini-2.5-Pro | 8.34 | 6.92 | 51.35 | 37 | 34 | 29.85 |
| Gemini-2.5-Flash | 7.68 | 7.08 | 49.87 | 34 | 29 | 25.30 |
| Claude-3.5 | 7.81 | 6.86 | **52.10** | 36 | 33 | 29.40 |
| Claude-4 | 8.22 | 7.14 | 45.25 | 31 | 29 | 26.10 |

Table 6: **Applying SWG to BEVBert.** We compare the vanilla BEVBert with the one applying our SWG. **Bold** indicates the best result. We report the relative percentage change compared to the baseline in parentheses. Red denotes improvement.

| Method | NE↓ | OSR↑ | SR↑ | SPL↑ |
|---|---|---|---|---|
| BEVBert | 4.57 | 67 | 59 | 50 |
| **BEVBert w/ SWG** | **4.37**(4.3%) | **70**(4.4%) | **61**(3.3%) | **53**(6.0%) |

models suggests that STRIDER's design amplifies the strengths of the underlying model without relying on a specific model.

For our primary experiments, we selected Qwen-VL-Max. This choice highlights STRIDER's model-agnostic design, which enables our method to work with any similar model and ensures that its results are driven by the strength of our approach, rather than reliance on a specific foundation model.

#### 4.3.4 Applying SWG to Fine-Tuned Models

We further assess the effectiveness of Structured Waypoint Guidance (SWG) in fine-tuned models. Specifically, we incorporate our SWG module into the BEVBert model by replacing its original waypoint prediction mechanism. This modification leads to improvements across all key evaluation metrics, as shown in Table 6.

Even in fine-tuned settings, SWG's ability to integrate environmental structure as a strong prior helps compensate for uncertainties in the navigation task. This further reinforces the versatility and effectiveness of SWG in different contexts, demonstrating that structured priors can be seamlessly integrated into existing models, thereby improving their robustness and reliability across a range of navigation tasks.

## 5 Conclusion

We presented STRIDER, a zero-shot VLN-CE framework that optimizes agent behavior through Instruction-Aligned Structural Decision Space Optimization. By combining a Structured Waypoint Generator with a Task-Alignment Regulator, STRIDER enables agents to navigate in complex, unseen environments in a manner that is both structurally coherent and semantically faithful to natural language instructions. Extensive experiments on VLN-CE benchmarks demonstrate that our approach outperforms strong zero-shot baselines across multiple metrics, with ablations confirming the complementary contributions of structural planning and feedback-driven regulation. This work highlights the importance of structuring and modulating the decision space for long-horizon instruction following and opens up future directions in integrating richer spatial priors, adaptive subgoal modeling, and instruction-aware exploration strategies.

## 6 Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 62293543, Grant 62322605.

## References

[1] Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. Bevbert: Multimodal map pre-training for language-guided navigation. *arXiv preprint arXiv:2212.04385*, 2022.

[2] Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[3] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. 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, 2018.

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

[5] Kehan Chen, Dong An, Yan Huang, Rongtao Xu, Yifei Su, Yonggen Ling, Ian Reid, and Liang Wang. Constraint-aware zero-shot vision-language navigation in continuous environments. *arXiv preprint arXiv:2412.10137*, 2024.

[6] Peihao Chen, Dongyu Ji, Kunyang Lin, Runhao Zeng, Thomas Li, Mingkui Tan, and Chuang Gan. Weakly-supervised multi-granularity map learning for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 35:38149–38161, 2022.

[7] Peihao Chen, Xinyu Sun, Hongyan Zhi, Runhao Zeng, Thomas H Li, Gaowen Liu, Mingkui Tan, and Chuang Gan. A$\hat{2}$ nav: Action-aware zero-shot robot navigation by exploiting vision-and-language ability of foundation models. *arXiv preprint arXiv:2308.07997*, 2023.

[8] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 34, pages 5834–5847, 2021.

[9] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan 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*, pages 16537–16547, 2022.

[10] Xinyi Chen, Boyu Zhou, Jiarong Lin, Yichen Zhang, Fu Zhang, and Shaojie Shen. Fast 3d sparse topological skeleton graph generation for mobile robot global planning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10283–10289. IEEE, 2022.

[11] Zihao Chen, Kunhong Li, Haoran Li, Zhiheng Fu, Hanmo Zhang, and Yulan Guo. Metric localization for lunar rovers via cross-view image matching. *Visual Intelligence*, 2(1):12, 2024.

[12] Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *International conference on machine learning*, pages 894–903. PMLR, 2017.

[13] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. *Advances in neural information processing systems*, 31, 2018.

[14] Yuanyuan Gao, Yalun Dai, Hao Li, Weicai Ye, Junyi Chen, Danpeng Chen, Dingwen Zhang, Tong He, Guofeng Zhang, and Junwei Han. Cosurfgs: Collaborative 3d surface gaussian splatting with distributed learning for large scene reconstruction. *arXiv preprint arXiv:2412.17612*, 2024.

[15] Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1634–1643, 2021.

[16] Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13137–13146, 2020.

[17] Yicong Hong, Cristian Rodriguez-Opazo, Qi Wu, and Stephen Gould. Sub-instruction aware vision-and-language navigation. *arXiv preprint arXiv:2004.02707*, 2020.

[18] Yicong Hong, Zun Wang, Qi Wu, and Stephen 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*, pages 15439–15449, 2022.

[19] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. A recurrent vision-and-language bert for navigation. *arXiv preprint arXiv:2011.13922*, 2020.

[20] Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. *arXiv preprint arXiv:1907.05446*, 2019.

[21] Seongjun Jeong, Gi-Cheon Kang, Joochan Kim, and Byoung-Tak Zhang. Zero-shot vision-and-language navigation with collision mitigation in continuous environment. *arXiv preprint arXiv:2410.17267*, 2024.

[22] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, 32(9-10):1194–1227, 2013.

[23] Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha Srinivasa. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6741–6749, 2019.

[24] Jacob Krantz, Aaron Gokaslan, Dhruv Batra, Stefan Lee, and Oleksandr Maksymets. Waypoint models for instruction-guided navigation in continuous environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15162–15171, 2021.

[25] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Proceedings of the European Conference on Computer Vision*, pages 104–120, 2020.

[26] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. *arXiv preprint arXiv:2010.07954*, 2020.

[27] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[28] Ta-Chih Lee, Rangasami L Kashyap, and Chong-Nam Chu. Building skeleton models via 3-d medial surface axis thinning algorithms. *CVGIP: graphical models and image processing*, 56(6):462–478, 1994.

[29] Hao Li, Yuanyuan Gao, Haosong Peng, Chenming Wu, Weicai Ye, Yufeng Zhan, Chen Zhao, Dingwen Zhang, Jingdong Wang, and Junwei Han. Dgtr: Distributed gaussian turbo-reconstruction for sparse-view vast scenes. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 207–213. IEEE, 2025.

[30] Hao Li, Yuanyuan Gao, Chenming Wu, Dingwen Zhang, Yalun Dai, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Junwei Han. Ggrt: Towards pose-free generalizable 3d gaussian splatting in real-time. In *European Conference on Computer Vision*, pages 325–341. Springer, 2024.

[31] Hao Li, Roy Qin, Zhengyu Zou, Diqi He, Bohan Li, Bingquan Dai, Dingewn Zhang, and Junwei Han. Langsurf: Language-embedded surface gaussians for 3d scene understanding. *arXiv preprint arXiv:2412.17635*, 2024.

[32] Hao Li, Dingwen Zhang, Yalun Dai, Nian Liu, Lechao Cheng, Jingfeng Li, Jingdong Wang, and Junwei Han. Gp-nerf: Generalized perception nerf for context-aware 3d scene understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21708–21718, 2024.

[33] Wenji Li, Zhaojun Wang, Ruitao Mai, Pengxiang Ren, Qinchang Zhang, Yutao Zhou, Ning Xu, JiaFan Zhuang, Bin Xin, Liang Gao, et al. Modular design automation of the morphologies, controllers, and vision systems for intelligent robots: a survey. *Visual Intelligence*, 1(1):2, 2023.

[34] Xiangyang Li, Zihan Wang, Jiahao Yang, Yaowei Wang, and Shuqiang Jiang. Kerm: Knowledge enhanced reasoning for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2583–2592, 2023.

[35] Xiwen Liang, Liang Ma, Shanshan Guo, Jianhua Han, Hang Xu, Shikui Ma, and Xiaodan Liang. Cornav: Autonomous agent with self-corrected planning for zero-shot vision-and-language navigation. *arXiv preprint arXiv:2306.10322*, 2023.

[36] Jyh-Ming Lien, John Keyser, and Nancy M Amato. Simultaneous shape decomposition and skeletonization. In *Proceedings of the 2006 ACM symposium on Solid and physical modeling*, pages 219–228, 2006.

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

[38] Yuxing Long, Xiaoqi Li, Wenzhe Cai, and Hao Dong. Discuss before moving: Visual language navigation via multi-expert discussions. In *Proceedings of the International Conference on Robotics and Automation*, pages 17380–17387, 2024.

[39] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. *arXiv preprint arXiv:1901.03035*, 2019.

[40] Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The regretful agent: Heuristic-aided navigation through progress estimation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 6732–6740, 2019.

[41] Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. In *Proceedings of the European Conference on Computer Vision*, pages 259–274, 2020.

[42] Yanyuan Qiao, Wenqi Lyu, Hui Wang, Zixu Wang, Zerui Li, Yuan Zhang, Mingkui Tan, and Qi Wu. Open-nav: Exploring zero-shot vision-and-language navigation in continuous environment with open-source llms. *arXiv preprint arXiv:2409.18794*, 2024.

[43] Nathan D Ratliff, David Silver, and J Andrew Bagnell. Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots*, 27:25–53, 2009.

[44] Sonia Raychaudhuri, Saim Wani, Shivansh Patel, Unnat Jain, and Angel X Chang. Language-aligned waypoint (law) supervision for vision-and-language navigation in continuous environments. *arXiv preprint arXiv:2109.15207*, 2021.

[45] Hossein Memarzadeh Sharifipour, Bardia Yousefi, and Xavier PV Maldague. Skeletonization and reconstruction based on graph morphological transformations. *arXiv preprint arXiv:2009.07970*, 2020.

[46] Xiangyu Shi, Zerui Li, Wenqi Lyu, Jiatong Xia, Feras Dayoub, Yanyuan Qiao, and Qi Wu. Smartway: Enhanced waypoint prediction and backtracking for zero-shot vision-and-language navigation. *arXiv preprint arXiv:2503.10069*, 2025.

[47] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3009, 2023.

[48] Yuliang Sun, Xudong Zhang, and Yongwei Miao. A review of point cloud segmentation for understanding 3d indoor scenes. *Visual Intelligence*, 2(1):14, 2024.

[49] Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, and Pieter Abbeel. Value iteration networks. *Advances in neural information processing systems*, 29, 2016.

[50] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. *arXiv preprint arXiv:1904.04195*, 2019.

[51] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406. PMLR, 2020.

[52] Marc Toussaint. Logic-geometric programming: An optimization-based approach to combined task and motion planning. In *IJCAI*, pages 1930–1936, 2015.

[53] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei 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*, pages 6629–6638, 2019.

[54] Yizhou Wang, Longguang Wang, Qingyong Hu, Yan Liu, Ye Zhang, and Yulan Guo. Panoptic segmentation of 3d point clouds with gaussian mixture model in outdoor scenes. *Visual Intelligence*, 2(1):10, 2024.

[55] Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, Junjie Hu, Ming Jiang, and Shuqiang Jiang. Lookahead exploration with neural radiance representation for continuous vision-language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13753–13762, 2024.

[56] Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Gridmm: Grid memory map for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15625–15636, 2023.

[57] Richard Wu. Recbert: Semantic recommendation engine with large language model enhanced query segmentation for k-nearest neighbors ranking retrieval. *Intelligent and Converged Networks*, 5(1):42–52, 2024.

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

[59] Lingfeng Zhang, Xiaoshuai Hao, Qinwen Xu, Qiang Zhang, Xinyao Zhang, Pengwei Wang, Jing Zhang, Zhongyuan Wang, Shanghang Zhang, and Renjing Xu. Mapnav: A novel memory representation via annotated semantic maps for vlm-based vision-and-language navigation. *arXiv preprint arXiv:2502.13451*, 2025.

[60] Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13624–13634, 2024.

[61] Gengze Zhou, Yicong Hong, and Qi Wu. 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, 2024.

[62] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10012–10022, 2020.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The key assertions presented in the abstract and introduction effectively encapsulate the research goals, methodologies, and contributions of this work, all of which are further developed and rigorously supported throughout the whole paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide the limitation discussion in the supplementary materials.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: Our paper is based on experimental results and does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed description of the method and experimental setup in the paper, to ensure the reproducibility of the methods.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the code in the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The experimental setting and details are provided in the Sec. 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: Our paper does not include error bars or other statistical experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The information is provided in the Sec. 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our research complies with the NeurIPS Code of Ethics in all aspects.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We provide this in the supplementary material.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The existing assets covered in the paper are subject to their license and terms of use, and are explicitly cited and described.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide the code in the supplementary material.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Please see Section 4.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.