SCALABLE NATIVE AGENTIC INTELLIGENCE FOR LONG-HORIZON VISION-AND-LANGUAGE NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-and-language navigation requires agents to interact with unknown environments over long horizons while continually coordinating perception, memory, planning, and action as observations evolve. Existing agentic approaches to VLN primarily rely on orchestrating multiple specialized models, each serving a different purpose. However, stitching together heterogeneous models introduces substantial pipeline complexity, along with unnecessary inference overhead and cost. In this paper, we present *SNAIL-VLN*, a native agentic framework in which a shared multimodal language model jointly learns active exploration, action prediction, spatial perception and memory, and task planning and scheduling, while a runtime harness composes these capabilities in a closed loop. To support these capabilities, we first construct a spatial-annotated training dataset from 153 indoor scenes, providing region-topology supervision and 4,742 DFS-guided exploratory expert trajectories for episode-level spatial memory and active exploration. Building upon this dataset, we curate five types of capability-specific question-answer pairs for joint training and design a harness that dynamically combines the shared model's outputs while maintaining execution state and interacting with the environment. Extensive experiments on LH-VLN show that our native agentic model establishes a new state of the art, improving the multi-stage task success rate from 2.44% to 11.40%. Module-by-module ablations further show that each atomic capability contributes to the overall performance. Finally, scaling experiments suggest that increasing model capacity generally improves navigation performance, with the clearest gains in multi-stage and subtask success. We will make our code, model checkpoints, and data publicly available to facilitate further research.

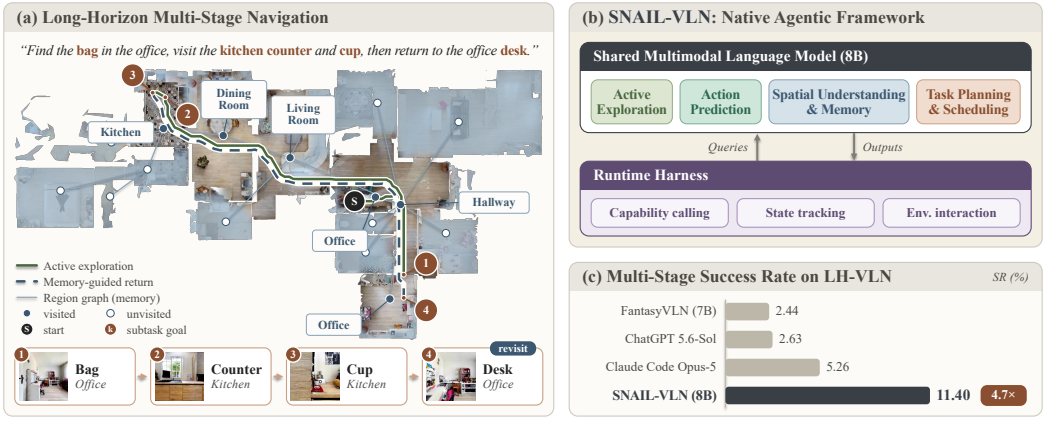


Figure 1: Overview of method. Unlike existing agentic systems relying on external models or tool orchestration, *SNAIL-VLN* natively exhibits agentic capabilities for long-horizon, multi-stage navigation, outperforming prior SOTA *FantasyVLN* and strong proprietary agentic models.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Vision-and-Language Navigation (VLN) is one of the fundamental tasks in embodied AI, which requires embodied agents to navigate complex, unseen environments guided by natural language instructions and visual observations (Anderson et al., 2018; Gu et al., 2022). Aligning dynamic visual streams with high-level language-based goals is essential for successful navigation (Anderson et al., 2018; Hao et al., 2020). Driven by the strong multimodal capabilities of large-scale pretrained vision-language models (VLMs), fine-tuning these models on offline expert navigation trajectories has emerged as a prevalent practice (Zheng et al., 2024; Zhou et al., 2024a).

Existing VLN approaches include end-to-end methods (Zheng et al., 2024) and reasoning-based methods (Zhou et al., 2024b; Zuo et al., 2026). End-to-end methods directly map inputs to actions for rapid decision-making, while reasoning-based methods leverage textual or multimodal Chain-of-Thought (CoT) to handle complex scenarios (Wei et al., 2022; Zhou et al., 2024b; Zuo et al., 2026). However, neither end-to-end nor reasoning-based methods are immune to error accumulation and context overload, especially in long-horizon, multi-stage tasks Song et al. (2025).

Motivated by recent breakthroughs in autonomous coding agents (Jimenez et al., 2024; Yang et al., 2024), agentic methods are attracting increasing attention in VLN (Li et al., 2026; Long et al., 2025; Xin et al., 2026). These methods utilize an external harness to assist the model in making reliable decisions over sustained, long-horizon interactions (Li et al., 2026; Long et al., 2025; Xin et al., 2026). Agentic approaches can separate semantic planning from low-level execution through explicit subgoal management (Long et al., 2025; Zhang et al., 2025b; Ma et al., 2026). By breaking down the navigation objective into discrete subgoals, these approaches organize execution around individual subtasks (Long et al., 2025; Zhang et al., 2025b; Ma et al., 2026). Furthermore, external tool invocation can support structured memory maintenance, while explicit feedback and self-correction mechanisms can mitigate cascading errors in long-horizon decision-making (Xin et al., 2026; Xue et al., 2026; He et al., 2025; Dai et al., 2026). However, these capabilities are typically distributed across multiple specialized models or modules, such as separate components for perception, planning, waypoint generation, and execution (Chen et al., 2025; Ma et al., 2026; Xin et al., 2026; Xue et al., 2026). To accommodate models with heterogeneous architectures, inference speeds, and environment dependencies, the agentic pipeline inevitably grows in complexity, compromising efficiency and increasing deployment cost. More fundamentally, whether a single model can learn to support these capabilities in a shared representation, while a harness dynamically composes them during execution rather than delegating each capability to a separate learned model, remains largely unexplored (Li et al., 2026; Ma et al., 2026; Xin et al., 2026; Xue et al., 2026; Dai et al., 2026).

We propose *SNAIL-VLN* to explore scalable native agentic intelligence for long-horizon VLN. It features a shared multimodal language model that jointly learns active exploration, action prediction, spatial perception and memory, and task planning and scheduling, together with a runtime harness that composes these capabilities in a closed loop. Rather than treating these abilities as isolated external services, we train them jointly as atomic capabilities, while the harness dynamically composes their outputs according to the current instruction, observations, and execution state, maintains spatial memory, and executes the resulting actions. This formulation preserves a shared vision-language representation while retaining the flexibility of closed-loop capability orchestration. To enable this design, we propose *SNAIL-Dataset*, a spatial-topology-annotated training dataset constructed from 153 indoor scenes. It includes region topology graphs and 4,742 DFS-guided exploratory expert trajectories, from which we derive capability-specific question-answer pairs for joint training. Extensive experiments on LH-VLN (Song et al., 2025) show that *SNAIL-VLN* establishes a new state of the art, improving the multi-stage task success rate from FantasyVLN's 2.44% (Zuo et al., 2026) to 11.40%. Module-by-module ablations verify that each atomic capability contributes to the final performance, while scaling experiments show overall gains as the model size increases.

Our contributions are summarized as follows: (i) We introduce a shared multimodal language model that jointly learns active exploration, action prediction, spatial perception and memory, and task planning and scheduling, together with a runtime harness that composes them during closed-loop navigation. (ii) We construct a spatial-topology-annotated training dataset with 153 indoor scenes and 4,742 DFS-guided exploratory trajectories, derive capability-specific supervision for joint model training, and implement dynamic capability composition through the runtime harness. (iii) We achieve state-of-the-art performance on LH-VLN, and provide component ablations and model-scaling analyses that evaluate the contribution and scalability of the proposed design.

2 Related Work

2.1 VLN Foundations and Long-Horizon Settings

Vision-and-Language Navigation (VLN) was originally formulated as instruction following over discrete viewpoints in realistic indoor environments, with Room-to-Room (R2R) establishing the standard task setting (Anderson et al., 2018). Subsequent benchmarks extend VLN along complementary dimensions. R4R (Jain et al., 2019) considers longer routes with greater emphasis on instruction fidelity and path alignment, while REVERIE (Qi et al., 2020) requires agents to navigate to and ground referred objects. VLN-CE moves beyond discrete graph-based actions to continuous control, introducing an additional gap between high-level semantic decisions and executable motion (Krantz et al., 2020). More recently, LH-VLN studies multi-stage, long-horizon navigation with substantially longer interaction sequences and explicit evaluation of task completion across stages (Song et al., 2025). These increasingly long and interactive settings require agents to maintain navigation history, ground spatial evidence, recover from intermediate errors, and coordinate decisions across stages, motivating models that integrate multiple capabilities over the course of an episode.

2.2 Pretraining, Reasoning, and Spatial Memory

A substantial body of work improves VLN through stronger multimodal representations and structured navigation states. PREVALENT (Hao et al., 2020) introduces image-text-action pretraining for transferable navigation representations, VLN-BERT (Hong et al., 2021) adapts recurrent vision-language pretraining to VLN, and HAMT (Chen et al., 2021) explicitly models navigation history for multimodal decision making. DUET (Chen et al., 2022) combines global graph reasoning with local action selection, incorporating explicit spatial structure into long-range navigation. More recent approaches leverage large language and vision-language models for explicit intermediate reasoning. NavGPT (Zhou et al., 2024b) and NavGPT-2 (Zhou et al., 2024a) introduce foundation-model-based planning and navigation reasoning; NavCoT (Lin et al., 2025) decomposes reasoning into prediction, matching, and action decision; and FantasyVLN (Zuo et al., 2026) combines textual and multimodal chain-of-thought supervision for long-horizon VLN. Complementary to explicit reasoning, memory-oriented methods construct compact representations of previously observed environments. MapNav (Zhang et al., 2025a), for example, replaces long visual histories with an incrementally updated semantic map. Collectively, these methods improve individual components of navigation, including representation, reasoning, and memory. Our work instead focuses on jointly learning multiple navigation capabilities within a shared representation and coordinating them during long-horizon interaction.

2.3 Agentic VLN: Planning, Skill Orchestration, and Active Feedback

Recent work increasingly approaches VLN as an agentic process that coordinates planning, perception, memory, and execution. InstructNav (Long et al., 2025) dynamically updates language-based plans and value maps, AO-Planner (Chen et al., 2025) combines foundation-model reasoning with affordance-guided waypoint and local-path planning, and CityNavAgent (Zhang et al., 2025b) performs hierarchical semantic planning with global topological memory. Other methods make capability orchestration explicit. SkillNav (Ma et al., 2026) routes subgoals to specialized navigation agents, while AgentVLN (Xin et al., 2026) employs a VLM as a high-level controller over perception, localization, planning, and execution skills. ProFocus (Xue et al., 2026) actively acquires task-relevant visual evidence to focus subsequent reasoning, whereas STRIDER (He et al., 2025) incorporates execution feedback into instruction-aligned structural decisions. These approaches demonstrate the value of coordinating complementary capabilities for long-horizon navigation. In many cases, however, capabilities are instantiated as separate modules, tools, or specialized agents and coordinated at the system level (Ma et al., 2026; Xin et al., 2026; Xue et al., 2026). In contrast, *SNAIL-VLN* jointly learns four navigation capabilities within a shared multimodal language model using five QA tasks, while a lightweight runtime harness coordinates capability invocation and maintains execution state. This design separates *capability learning* within a shared model from *capability orchestration* at runtime, enabling closed-loop agentic navigation without assigning each capability to a separate learned expert.

3 NATIVE AGENTIC MODEL

3.1 PROBLEM FORMULATION

We formulate VLN as a partially observable Markov decision process (POMDP) (Kaelbling et al., 1998). Let π denote an embodied agent, $\mathcal{E}$ the environment, $\mathcal{I}$ a natural-language instruction, and $\mathcal{A}$ the action space. The agent π is initialized at a randomly sampled starting waypoint and is tasked with navigating to the target specified by $\mathcal{I}$. At each timestep t, the agent receives a multi-view visual observation o_t from the environment and predicts an action chunk $a_t \in \mathcal{A}$ conditioned on $\mathcal{I}$ and the observation history $o_{\leq t}$. The predicted action chunk is then executed in the environment, yielding the next observation o_{t+1}. This process continues until the agent issues a stop action or reaches the maximum number of allowed action steps.

VLN poses three key challenges. First, observations are inherently incomplete, especially at early stages, requiring the agent to actively explore the environment under uncertainty (Krantz et al., 2020; Long et al., 2025). Second, as navigation proceeds, the growing observation history introduces increasingly long and redundant visual context, requiring the agent to abstract observations into compact spatial memory and efficiently retrieve relevant information (Song et al., 2025; Zhang et al., 2025a; Li et al., 2026). Finally, long-horizon tasks often involve multiple interdependent stages, requiring the agent to plan and manage context across stages for coherent execution (Song et al., 2025; Zhang et al., 2025b).

3.2 OVERVIEW

To address these challenges, we propose *SNAIL-VLN*, which consists of a multimodal language model π_θ and a runtime harness $\mathcal{H}$. The model π_θ learns four atomic capabilities within a shared representation space: active exploration, action prediction, spatial understanding and memory, and task planning and scheduling. The harness $\mathcal{H}$ orchestrates these capabilities and maintains the execution state at runtime. By internalizing these capabilities within a single model, *SNAIL-VLN* enables native agentic navigation without relying on external models or tool calls.

We construct spatially grounded supervision to acquire these capabilities. Specifically, we annotate region-level topology over indoor environments and generate DFS-guided exploratory trajectories. From these annotations and trajectories, we construct five QA supervision tasks: navigation decision, subtask decomposition, region identification, target visibility judgment, and same-region judgment. We train *SNAIL-VLN* end-to-end across all supervision tasks with a causal language-modeling loss, enabling scalable and efficient learning.

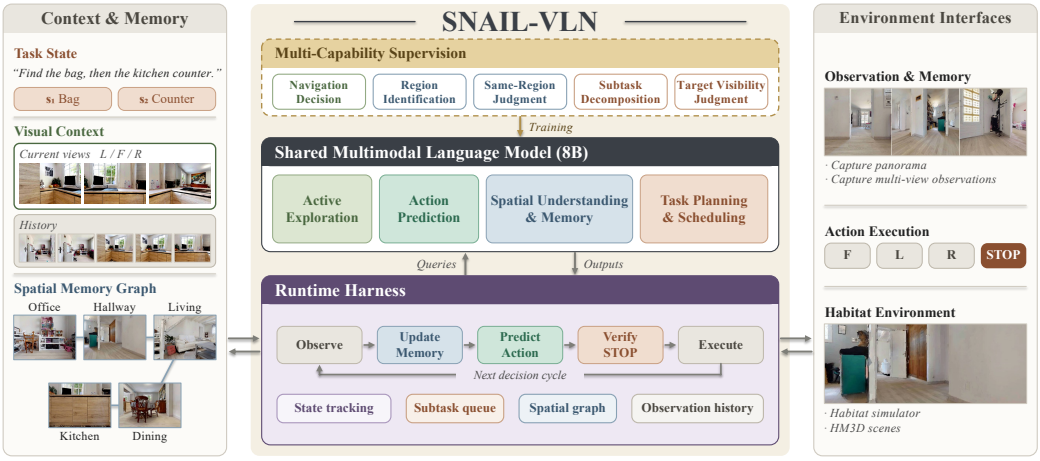


Figure 2: *SNAIL-VLN*: a shared multimodal language model jointly trained on five QA tasks for four atomic capabilities, composed at test time by a closed-loop runtime harness that tracks execution state and interacts with the environment.

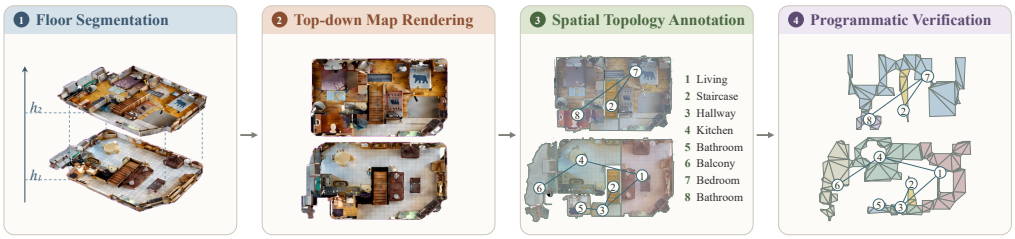


Figure 3: Spatial-topology data construction, from which we derive region graphs and trajectories.

3.3 Spatial-Topology Data Construction

We construct spatially grounded training data from 153 HM3D indoor scenes (Ramakrishnan et al., 2021) using Habitat (Savva et al., 2019). For each scene, we segment the floor layout, render a top-down map, manually annotate semantic regions, and programmatically verify the resulting annotations. We represent each environment as a region-level topology graph, where nodes denote semantic regions and edges denote traversable connectivity.

To provide supervision beyond shortest-path navigation, we generate 4,742 exploratory expert trajectories over the topology graphs, guided by depth-first search (DFS) (Tarjan, 1972). Each trajectory starts from the task start location and prioritizes adjacent unexplored regions. When multiple candidates are available, we first prefer the region with a smaller angular deviation from the agent's current heading, and then the one with a shorter distance. The trajectory terminates once the task endpoint is reached. DFS is used only to construct training trajectories and does not constrain the policy at test time. We further discard trajectories whose endpoint snap point does not provide a visible target object, as such trajectories do not yield reliable completion supervision.

From the resulting trajectories and spatial annotations, we construct five QA supervision tasks: navigation decision, subtask decomposition, region identification, target visibility judgment, and same-region judgment.

3.4 Native Multi-Capability Learning

We train all four capabilities within the same multimodal language model π_θ using the five QA supervision tasks constructed above. Navigation-decision supervision supports active exploration and action prediction, region identification and same-region judgment support spatial understanding and memory, and subtask decomposition and target visibility judgment support task planning and scheduling.

Active Exploration and Action Prediction. We slice the DFS-guided expert trajectories to construct local policy supervision for active exploration and action prediction. Each example uses the current subtask and visual-spatial context as input, and the subsequent trajectory segment as the target action chunk. Since DFS trajectories may traverse unexplored regions before reaching the subtask target, they provide supervision for exploratory behavior within the same task-conditioned policy. For the current subtask s_k, the model predicts an action chunk as

$$a_t \sim \pi_\theta\left(\cdot \mid s_k,\, o_t,\, \mathcal{O}_t,\, \mathrm{Text}(G_t)\right), \qquad a_t \in \mathcal{A}. \tag{1}$$

Here, G_t provides the accumulated spatial context for action prediction.

Spatial Understanding and Memory. We use region identification and same-region judgment to supervise spatial understanding and memory. The model identifies the current semantic region from the current observation and recent history, and determines whether the current and stored panoramas correspond to the same region. The two predictions are

$$\begin{aligned} r_t &\sim \pi_\theta\left(\cdot \mid o_t,\, \mathcal{O}_t\right), \\ m_{t,v} &\sim \pi_\theta\left(\cdot \mid p_t,\, p_v\right), \qquad m_{t,v} \in \{0,1\}, \end{aligned} \tag{2}$$

where r_t denotes the current region category, p_t the current panorama, and p_v the panorama stored at node v. These predictions provide the perceptual signals required by $\mathcal{H}$ to construct and update the spatial memory.

Task Planning and Scheduling. We supervise task planning and scheduling in long-horizon, multi-stage navigation with subtask decomposition and target visibility judgment. The model decomposes the instruction into an ordered sequence of single-stage subtasks and verifies whether the target of the current subtask is visible from recent observations. The corresponding predictions are

$$\begin{aligned} S &\sim \pi_\theta\left(\cdot \mid \mathcal{I}\right), \qquad S = [s_1, \ldots, s_K], \\ z_t &\sim \pi_\theta\left(\cdot \mid s_k,\, \mathcal{O}_{t-4:t}\right), \qquad z_t \in \{0, 1\}, \end{aligned} \tag{3}$$

where s_k denotes the current subtask and $z_t = 1$ indicates that its target is visually confirmed. These predictions provide the task structure and completion signals required by $\mathcal{H}$ for runtime scheduling.

Training Objective. We train all five QA tasks end-to-end within the same multimodal language model. For a training example (x, y), where x denotes the task-specific multimodal input and y the target response, we optimize the standard causal language-modeling objective (Zheng et al., 2024)

$$\mathcal{L}(\theta) = -\sum_{j=1}^{|y|} \log p_\theta\left(y_j \mid y_{<j},\, x\right). \tag{4}$$

The five supervision sources are mixed through repeat sampling during training. This unified formulation allows the agentic capabilities to benefit from the established scalability of language-model training (Kaplan et al., 2020) without introducing capability-specific models or objectives.

3.5 Runtime Harness

Given a long-horizon, multi-stage instruction $\mathcal{I}$, π_θ first decomposes it into an ordered sequence of single-stage subtasks. The harness $\mathcal{H}$ tracks task progress and selects the current unfinished subtask for execution. The subtasks are executed sequentially, while the accumulated spatial memory is retained across stages to support subsequent navigation and planning.

For each subtask, $\mathcal{H}$ drives navigation through a sequence of decision cycles, each consisting of a spatial memory update followed by action prediction and execution. At the beginning of each cycle, π_θ identifies the current region from the multi-view observation and recent history, while $\mathcal{H}$ captures a panoramic observation for spatial matching. Based on the predicted region category, $\mathcal{H}$ retrieves candidate nodes of the same category from the spatial graph and queries π_θ to compare the current panorama with the panorama stored at each candidate node and determine whether they correspond to the same region. If a match is identified, $\mathcal{H}$ marks the corresponding node as the current location and updates its visitation state; otherwise, it creates a new node and connects it to the previously visited node. The resulting spatial graph provides a compact abstraction of the explored environment and is retained across subtasks. After updating the spatial memory, $\mathcal{H}$ constructs the decision context from the current subtask, multi-view observation, recent history, and a textual representation of the spatial graph. Conditioned on this context, π_θ predicts an action chunk, which is executed by $\mathcal{H}$.

When the predicted action chunk contains `STOP`, $\mathcal{H}$ queries π_θ to verify the visibility of the subtask target from recent observations. If the verification fails, `STOP` is removed and the remaining actions are executed. Otherwise, the complete action chunk, including `STOP`, is sent to the environment, and $\mathcal{H}$ advances to the next subtask after the stop action is executed. After each environment interaction, $\mathcal{H}$ updates the observation history and execution state, and proceeds until all subtasks are completed. The overall runtime procedure of $\mathcal{H}$ is summarized in Algorithm 1.

4 Experiments

We evaluate long-horizon navigation performance, the effects of exploratory supervision and runtime components, model scaling, and inference efficiency.

4.1 Experimental Setup

Benchmark and Data. We evaluate on LH-VLN (Song et al., 2025), where each episode requires sequential completion of multiple navigation subtasks. We use the official scene-based split: scene IDs $\leq$ 700 for training, 701–800 for validation, and $>$ 800 for testing. Following the endpoint-visibility criterion in Section 3.3, we retain episodes only when the target is visible at every subtask endpoint, yielding 411 training, 54 validation, and 114 test episodes. We scale up the training episodes from 411 to 4,742 and render the corresponding DFS-guided exploratory trajectories. From these trajectories and spatial-topology annotations of HM3D (Ramakrishnan et al., 2021) training scenes, we construct 1,076,791 QA training samples.

Metrics. Following LH-VLN (Song et al., 2025), we report multi-stage task success rate (SR), which requires completing the full episode, and independent subtask success rate (ISR), which evaluates each subtask independently of the others. Conditional success rate (CSR) weights subtask success by the success of preceding subtasks. CSR weighted by Ground Truth (CGT) further weights CSR by ground-truth trajectory length to account for trajectory difficulty. All four metrics are percentages; higher is better.

Baselines. We compare three groups of baselines under the LH-VLN evaluation protocol. **End-to-end** baselines include Random (Song et al., 2025), NaviLLM (Pretrain), and NaviLLM (Fine-tuned) (Zheng et al., 2024). **Reasoning-based** baselines include GLM-4V Prompt (Team GLM, 2024), GPT-4 + NaviLLM and MGDM (Song et al., 2025), Aux-Think (Wang et al., 2025a), and

Algorithm 1 SNAIL-VLN Runtime Harness.

```
# I, E, pi: instruction, environment, and multimodal language model
# S = [s_1, ..., s_K]: ordered subtask list
# G, O: spatial memory graph and observation history
# o_t, r_t, p_t: observation, region, and panorama at step t
# v_t: matched node in G, or None
# a_t = [...]: predicted action chunk

S = subtask_decompose(I, pi)
s_k = pop_front(S)
G, O = empty, []

while not terminated(E):
    o_t = observe(E)

    # update spatial memory
    r_t, p_t = region_recognize(o_t, O, pi), capture_panorama(E)
    v_t = match_region(p_t, r_t, G, pi)
    G = update_graph(G, v_t, r_t, p_t)

    # predict action chunk
    a_t = action_predict(s_k, o_t, O, text(G), pi)

    # verify STOP before execution
    if STOP in a_t:
        visible = target_visible(s_k, O[-5:], pi)
        if not visible:
            a_t = remove_stop(a_t)

    # execute the entire action chunk
    o_next = execute(E, a_t)
    O = update_history(O, o_t, a_t, o_next)

    # advance only after STOP is executed by the environment
    if STOP in a_t and not terminated(E):
        s_k = pop_front(S)
```

Table 1: Navigation performance on LH-VLN.

Method	SR ↑	ISR ↑	CSR ↑	CGT ↑
End-to-end				
Random	0	0	0	0
NaviLLM (Pretrain)	0	0	0	0
NaviLLM (Finetuned)	0	1.77	1.27	2.62
Reasoning-based				
GLM-4V Prompt	0	0	0	0
GPT-4 + NaviLLM	0	2.19	1.46	2.62
MGDM	0	2.35	1.65	2.92
Aux-Think	0.65	3.16	2.04	1.47
FantasyVLN	2.44	11.01	9.64	8.99
Agentic				
ChatGPT 5.6-Sol	2.63	14.29	10.58	9.68
Claude Code Opus-5	5.26	26.87	17.82	17.84
SNAIL-VLN	**11.40**	**32.99**	**26.16**	**27.26**

Table 2: Effect of training trajectory construction.

Trajectory	SR ↑	ISR ↑	CSR ↑	CGT ↑
Shortest Path	1.82	6.52	5.66	4.06
DFS	3.64	10.87	9.65	8.63

Table 3: Runtime component ablations. SD: subtask decomposition; SM: spatial memory; TV: target visibility judgment. ✓/× denotes enabled/disabled.

SD	SM	TV	SR ↑	ISR ↑	CSR ↑	CGT ↑
×	✓	✓	0.00	14.97	11.14	12.88
✓	×	✓	4.39	22.45	16.11	16.06
✓	✓	×	7.89	30.27	21.78	21.44
✓	✓	✓	**11.40**	**32.99**	**26.16**	**27.26**

Table 4: Model scaling within the InternVL3.5 family.

Size	SR ↑	ISR ↑	CSR ↑	CGT ↑
8B	**8.77**	**26.53**	**20.38**	20.83
4B	7.02	25.85	19.54	**20.87**
2B	6.14	21.43	17.74	19.11
1B	4.39	18.37	14.44	14.56

FantasyVLN (Zuo et al., 2026). **Agentic** baselines include ChatGPT 5.6-Sol (OpenAI, 2026) and Claude Code Opus-5 (Anthropic, 2026).

4.2 Main Results

As shown in Table 1, *SNAIL-VLN* achieves the best results across all four metrics, with 11.40% SR, 32.99% ISR, 26.16% CSR, and 27.26% CGT. These improvements highlight the benefit of jointly learning atomic agentic capabilities in a shared representation space, together with a runtime harness that manages memory updates, action execution, and subtask transitions. More broadly, agentic methods consistently outperform end-to-end and reasoning-based baselines, indicating that explicit planning, coordinated execution, and feedback are important for sustained progress in long-horizon, multi-stage tasks. Among non-agentic methods, FantasyVLN performs best overall, achieving 2.44% SR, 11.01% ISR, 9.64% CSR, and 8.99% CGT, compared with 0%, 1.77%, 1.27%, and 2.62%, respectively, for fine-tuned NaviLLM, the strongest end-to-end baseline. FantasyVLN unifies multiple CoT reasoning modes to improve the robustness of navigation decisions (Zuo et al., 2026). Nevertheless, the CoT-based semantic planning primarily operates at the local decision level, limiting coherent progress over long-horizon interactions.

4.3 Ablation Studies

Exploratory Supervision. We investigate whether exploratory supervision benefits navigation policy learning in partially observable environments. To this end, we compare shortest-path and DFS-guided trajectories constructed from the same 411 filtered training episodes, while keeping all other training and evaluation settings fixed. DFS-guided trajectories are generated through depth-first exploration of unvisited regions, whereas shortest-path trajectories follow globally shortest routes computed with full environment knowledge. As shown in Table 2, DFS-guided supervision improves SR, ISR, CSR, and CGT by 1.82, 4.35, 3.99, and 4.57 percentage points, respectively. When navigating unknown environments, agents need sufficient exploration before making informed navigation decisions (Long et al., 2025; Li et al., 2026). These results show that DFS-guided trajectories provide effective supervision for learning such exploratory behavior.

Runtime Component Ablations. Using the same full-model checkpoint, we disable subtask decomposition, spatial memory, and target visibility judgment individually while keeping all other

components unchanged. These ablations isolate the effect of using each learned capability at runtime, rather than its contribution as a training objective. As shown in Table 3, disabling subtask decomposition causes the largest degradation, reducing SR from 11.40% to 0.00%, while ISR remains at 14.97%. This gap indicates that the agent can still complete individual subtasks but struggles to organize them toward overall task completion without explicit decomposition. Removing spatial memory reduces SR to 4.39%, showing the importance of accumulated spatial context for navigation across extended interactions. Disabling target visibility judgment further reduces SR to 7.89%, indicating that reliable visibility assessment is important for determining when to terminate navigation. Together, these results demonstrate that subtask decomposition, spatial memory, and target visibility judgment provide complementary runtime capabilities for long-horizon task execution.

4.4 Scaling Analysis

To assess the effect of model capacity on multi-capability learning, we train InternVL3.5 (Wang et al., 2025b) backbones ranging from 1B to 8B parameters using the same 4,742 training trajectories, number of epochs, and optimization protocol. As shown in Table 4, SR, ISR, and CSR improve consistently with model size. Scaling from 1B to 8B increases SR from 4.39% to 8.77%, ISR from 18.37% to 26.53%, and CSR from 14.44% to 20.38%. CGT follows a similar overall trend, with the 4B and 8B models achieving comparable performance 20.87% and 20.83%, respectively. These results indicate that increased model capacity generally benefits the joint learning of multiple agentic capabilities, leading to stronger task and subtask completion.

4.5 Inference Efficiency

Closed-loop navigation requires frequent model calls, making inference efficiency important for responsive agent execution. We measure request-level efficiency by aggregating model requests from completed LH-VLN test episodes. For each request i, let n_i, ℓ_i, and τ_i denote the number of output tokens, latency, and time to first token (TTFT), respectively, where output tokens include thinking tokens. We report mean latency and TTFT, together with end-to-end throughput $\sum_i n_i / \sum_i \ell_i$ and post-TTFT throughput $\sum_i (n_i - 1) / \sum_i (\ell_i - \tau_i)$ in tokens per second (TPS). As shown in Table 5, *SNAIL-VLN* achieves the lowest mean latency of 0.983 s and TTFT of 0.560 s, while generating only 8.39 output tokens per request on average. It consequently achieves the highest end-to-end throughput at 8.53 TPS, despite the higher post-TTFT throughput of the external models. This result highlights the importance of both short responses and low initial latency for efficient closed-loop interaction, where frequent model calls can amplify per-request overhead. These measurements reflect request-level inference efficiency rather than full-episode execution time, and differences in hardware and serving configurations prevent direct attribution to model architecture alone.

Table 5: Request-level inference efficiency on LH-VLN.

Model	Avg. Latency (s)	Avg. TTFT (s)	Avg. Output Tokens	End-to-End TPS	Post-TTFT TPS
ChatGPT 5.6-Sol	19.497	17.041	88.75	4.75	52.77
Claude Code Opus-5	72.121	71.749	127.92	1.77	341.15
SNAIL-VLN	**0.983**	**0.560**	**8.39**	**8.53**	17.44

5 Conclusion

We have explored whether a single VLN model can natively exhibit agentic capabilities without relying on external models or tool calls. We find that this formulation benefits from shared representations across capabilities and low inference latency through native capability invocation. The proposed *SNAIL-VLN* demonstrates stronger long-horizon navigation performance than closed-source agentic models, including ChatGPT Sol 5.6 and Claude Code Opus 5. Built upon our large-scale spatial-topology-annotated VLN data, *SNAIL-VLN* further demonstrates promising scalability with increasing model capacity.

AI USE STATEMENT

In preparing this manuscript, we used generative AI tools only for writing refinement, including grammar, clarity, and presentation, and for literature searches to identify relevant references. These tools were not used to develop research ideas or methods, design or conduct experiments, or analyze or interpret results. The authors are responsible for verifying AI-assisted edits and references and take full responsibility for the accuracy, originality, and integrity of the final manuscript.

REFERENCES

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*, pp. 3674–3683, 2018.

Anthropic. Introducing Claude Opus 5, 2026. URL https://www.anthropic.com/news/claude-opus-5.

Jiaqi Chen, Bingqian Lin, Xinmin Liu, Lin Ma, Xiaodan Liang, and Kwan-Yee K. Wong. Affordances-oriented planning using foundation models for continuous vision-language navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23568–23576, 2025. doi: 10.1609/aaai.v39i22.34526.

Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/2e5c2cb8d13e8fba78d95211440ba326-Abstract.html.

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*, pp. 16537–16547, 2022.

Guangzhao Dai, Shuo Wang, Zihan Wang, Guo-Sen Xie, Yang Yang, Jinshan Pan, Qianru Sun, and Xiangbo Shu. History to future: Evolving agent with experience and thought for zero-shot vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15177–15187, 2026.

Jing Gu, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Wang. Vision-and-language navigation: A survey of tasks, methods, and future directions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 7606–7623, 2022. doi: 10.18653/v1/2022.acl-long.524.

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*, pp. 13134–13143, 2020. doi: 10.1109/CVPR42600.2020.01315.

Diqi He, Xuehao Gao, Hao Li, Junwei Han, and Dingwen Zhang. STRIDER: Navigation via instruction-aligned structural decision space optimization. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/26492c1e75070f6726e3d17f1db28e8f-Abstract-Conference.html.

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. VLN BERT: A recurrent vision-and-language BERT for navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1643–1653, 2021.

Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. Stay on the path: Instruction fidelity in vision-and-language navigation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1862–1872. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1181.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998. doi: 10.1016/S0004-3702(98)00023-X.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. URL https://arxiv.org/abs/2001.08361.

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

Yijian Li, Changze Li, Hantian Shi, Jiaying Luo, Jiyuan Cai, Ming Yang, and Tong Qin. AgenticNav: Zero-shot vision-and-language navigation as a tool-calling harness. *arXiv preprint arXiv:2606.10577*, 2026.

Bingqian Lin, Yunshuang Nie, Ziming Wei, Jiaqi Chen, Shikui Ma, Jianhua Han, Hang Xu, Xiaojun Chang, and Xiaodan Liang. NavCoT: Boosting LLM-based vision-and-language navigation via learning disentangled reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7):5945–5957, 2025. doi: 10.1109/TPAMI.2025.3554559.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. InstructNav: Zero-shot system for generic instruction navigation in unexplored environment. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2049–2060, 2025.

Tianyi Ma, Yue Zhang, Zehao Wang, and Parisa Kordjamshidi. Breaking down and building up: Mixture of skill-based vision-and-language navigation agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 13035–13065, 2026. doi: 10.18653/v1/2026.acl-long.595.

OpenAI. GPT-5.6 System Card, 2026. URL https://deploymentsafety.openai.com/gpt-5-6.

Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton 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*, pp. 9979–9988, 2020. doi: 10.1109/CVPR42600.2020.01000.

Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M. Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3D): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL https://arxiv.org/abs/2109.08238.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied AI research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9339–9347, 2019. URL https://arxiv.org/abs/1904.01201.

Xinshuai Song, Weixing Chen, Yang Liu, Weikai Chen, Guanbin Li, and Liang Lin. Towards long-horizon vision-language navigation: Platform, benchmark and method. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12078–12088, 2025.

Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2): 146–160, 1972. doi: 10.1137/0201010.

Team GLM. ChatGLM: A family of large language models from GLM-130B to GLM-4 All Tools. *arXiv preprint arXiv:2406.12793*, 2024. URL https://arxiv.org/abs/2406.12793.

Shuo Wang, Yongcai Wang, Wanting Li, Xudong Cai, Yucheng Wang, Maiyue Chen, Kaihui Wang, Zhizhong Su, Deying Li, and Zhaoxin Fan. Aux-Think: Exploring reasoning strategies for data-efficient vision-language navigation. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. URL https://arxiv.org/abs/2505.11886.

Weiyun Wang, Zhangwei Gao, Lixin Gu, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025b. URL https://arxiv.org/abs/2508.18265.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Zihao Xin, Wentong Li, Yixuan Jiang, Ziyuan Huang, Bin Wang, Piji Li, Jianke Zhu, Jie Qin, and Shengjun Huang. Agentvln: Towards agentic vision-and-language navigation. In *European Conference on Computer Vision*, pp. 114–132. Springer, 2026.

Wei Xue, Mingcheng Li, Xuecheng Wu, Jingqun Tang, Dingkang Yang, and Lihua Zhang. ProFocus: Proactive perception and focused reasoning in vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18129–18139, 2026.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Lingfeng Zhang, Xiaoshuai Hao, Qinwen Xu, Qiang Zhang, Xinyao Zhang, Pengwei Wang, Jing Zhang, Zhongyuan Wang, Shanghang Zhang, and Renjing Xu. MapNav: A novel memory representation via annotated semantic maps for VLM-based vision-and-language navigation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13032–13056. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.acl-long.638.

Weichen Zhang, Chen Gao, Shiquan Yu, Ruiying Peng, Baining Zhao, Qian Zhang, Jinqiang Cui, Xinlei Chen, and Yong Li. CityNavAgent: Aerial vision-and-language navigation with hierarchical semantic planning and global memory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 31292–31309, 2025b. doi: 10.18653/v1/2025.acl-long.1511.

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*, pp. 13624–13634, 2024. URL https://arxiv.org/abs/2312.02010.

Gengze Zhou, Yicong Hong, Zun Wang, Xin Eric Wang, and Qi Wu. NavGPT-2: Unleashing navigational reasoning capability for large vision-language models. In *European Conference on Computer Vision*, pp. 260–278. Springer, 2024a.

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, pp. 7641–7649, 2024b. doi: 10.1609/aaai.v38i7.28597.

Jing Zuo, Lingzhou Mu, Fan Jiang, Chengcheng Ma, Mu Xu, and Yonggang Qi. FantasyVLN: Unified multimodal chain-of-thought reasoning for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29674–29683, 2026.