A RECURSIVE SELF-IMPROVING HARNESS FOR TRAINING-FREE GENERALIST NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generalist embodied navigation requires agents to adapt their behavior as task objectives, spatial contexts, and environmental conditions evolve. While recent agentic navigation methods leverage pretrained vision-language models (VLMs) for flexible reasoning, they remain largely task-specific and lack mechanisms for reusing experience across tasks and environments. We introduce RING-Nav, a recursive self-improving (RSI) harness for training-free generalist navigation. RING-Nav maintains a contextualized navigation state (CNS) that unifies task progress, spatial evidence, episodic memory, and tool feedback, enabling a frozen VLM to dynamically orchestrate shared navigation tools. To support continuous improvement across episodes, an RSI experience graph (RSI-EG) records contextualized tool transitions and their outcomes, retrieves relevant past experience, and guides subsequent tool decisions without updating model parameters. RING-Nav thus unifies spatially grounded tool orchestration (SGTO) with experience-driven self-improvement, enabling a single navigation agent to adapt across diverse tasks and environments. Extensive experiments demonstrate state-of-the-art performance on both individual navigation tasks and heterogeneous multi-stage instructions.

1 INTRODUCTION

Generalist navigation is a fundamental capability of open-world embodied agents: agents need to autonomously perform heterogeneous tasks within a physical space, such as understanding, perception, planning, exploration, and interaction (Gao et al., 2024; Wang et al., 2026; Xiong et al., 2025). Unlike task-specific navigation, where the objective and execution strategy are largely fixed, open-world agents must adapt their behavior as task objectives, spatial contexts, and environmental conditions change over time. Existing approaches either rely on large-scale end-to-end training (Zhang et al., 2025c; 2026; Xue et al., 2026; Zhu et al., 2025) or training-free modular systems (Yin et al., 2025a; Long et al., 2025). The former provides strong task integration but incurs substantial training and adaptation costs, whereas the latter offers greater flexibility and zero-shot generalization but is susceptible to accumulated errors from independently designed modules.

An emerging alternative is to equip pretrained VLMs with an *agentic harness* (Mokaria et al., 2026), an external layer that augments the model with planning (Yao et al., 2023; Wang et al., 2023), memory (Park et al., 2023), skills (Wang et al., 2024), and feedback (Shinn et al., 2023; Madaan et al., 2023). Beyond manually designed harnesses, recursive self-improving (RSI) has recently emerged as a paradigm for autonomously refining agent behavior and tool orchestration (Gao et al., 2026b), with promising results in reasoning, coding, and multimodal tasks (Yin et al., 2025b; Zhang et al., 2025b; Liu et al., 2026). Recent studies have extended agentic harnesses to visual navigation (Dai et al., 2026; Xin et al., 2026), but primarily focus on individual tasks.

This limitation becomes particularly pronounced in heterogeneous embodied navigation, where tool decisions must adapt to partial, evolving spatial observations and changing task objectives. For example, finding an unseen object may require exploration, whereas approaching a localized target calls for goal-directed navigation. These decisions are also temporally coupled, as earlier actions alter subsequent spatial context and available information, while similar situations may recur across tasks and environments. Thus, generalist navigation requires spatially grounded tool orchestration that can reuse experience across episodes. This raises a central question:

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.

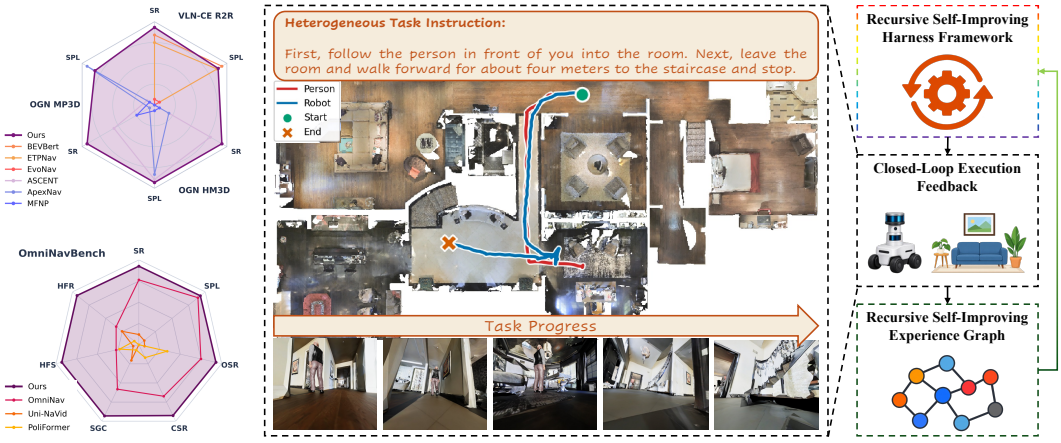


Figure 1: RING-Nav is a recursive self-improving harness for training-free heterogeneous generalist navigation. The framework executes heterogeneous instructions by updating the navigation context with execution feedback and leveraging cross-episode experience from the recursive self-improving experience graph to guide subsequent tool decisions. RING-Nav achieves strong performance across VLN-CE, OGN, and OmniNavBench.

How can an embodied agent continuously improve its tool decisions across heterogeneous tasks and environments without retraining the underlying model?

To address this question, we propose **RING-Nav**, an **RSI** har**N**ess for training-free heterogeneous **G**eneralist **Nav**igation. RING-Nav represents the evolving task, spatial, and execution context as a *contextualized navigation state* (CNS), which conditions the dynamic composition of navigation tools for objectives including instruction following, exploration, object search, and human interaction. To enable continual improvement, we introduce a cross-episode *recursive self-improving experience graph* (RSI-EG), which records tool transitions, their task and spatial contexts, and execution outcomes. At each decision point, relevant experience is retrieved from the graph to refine tool selection and transition strategies, enabling the harness to improve through accumulated experience without updating the underlying navigation models.

To the best of our knowledge, RING-Nav is the first RSI harness designed to enable heterogeneous generalist embodied navigation across diverse tasks and environments without task-specific model training. We evaluate RING-Nav across three complementary settings: language-guided visual navigation in VLN-CE (Krantz et al., 2020), ObjectNav (Batra et al., 2020), and heterogeneous multi-stage navigation in OmniNavBench (Sun et al., 2026). RING-Nav improves success rate over EvoNav (Dai et al., 2026) by 19.0% on R2R-CE and over ASCENT (Gong et al., 2026) by 1.6%/4.0% on HM3D/MP3D ObjectNav. On the heterogeneous OmniNavBench, RING-Nav further improves success rate from 23.75% to 28.75% and subgoal completion from 43.91% to 57.33% over OmniNav (Xue et al., 2026). Our main contributions are summarized as follows:

- We introduce **RING-Nav**, an RSI harness for training-free generalist navigation, enabling a unified agent to coordinate heterogeneous navigation objectives without updating model parameters.
- We propose spatially grounded tool orchestration (SGTO) that represents task progress, multimodal spatial evidence, episodic memory, and execution feedback in a unified contextualized navigation state (CNS), allowing tool decisions to adapt to evolving task and environmental contexts.
- We develop an RSI experience graph (RSI-EG) that converts cross-episode navigation experience into reusable tool-decision knowledge, enabling retrieval-guided adaptation and continuous improvement across tasks and environments without model retraining.

2 Related Works

Language-guided Visual Navigation spans diverse embodied navigation tasks with different goals and interaction requirements. Vision-and-Language Navigation in Continuous Environments (VLN-

CE) requires agents to follow natural-language instructions without predefined navigation graphs, making continuous-space waypoint selection and path planning essential (Hao et al., 2020; Chen et al., 2022; Krantz et al., 2020). Training-based methods learn navigation behaviors from trajectory supervision (Zhang et al., 2024; Wei et al., 2026), whereas zero-shot methods leverage pretrained foundation models without navigation-specific training (Shi et al., 2025; Qiao et al., 2025; Dai et al., 2026). Zero-shot Object Goal Navigation (ZS-OGN) emphasizes efficient exploration and semantic reasoning to locate unseen target objects (Majumdar et al., 2022; Yokoyama et al., 2024; Yin et al., 2024; Zhang et al., 2025e; Gong et al., 2026). Social Navigation (SocialNav) further considers human-populated environments and requires socially compliant motion around people (Song et al., 2025; Munje et al., 2025; Chen et al., 2026). Generalist navigation extends these settings toward cross-task navigation with a unified agent (Zhang et al., 2025c; Long et al., 2025; Zhang et al., 2026; Xue et al., 2026; Gao et al., 2026a).

Physical-World Agents combine foundation-model reasoning with perception and executable robot skills to interact with real environments. Early approaches employed LLMs to select, refine, and compose robot skills from language instructions, with applications primarily focused on manipulation (Ichter et al., 2023; Huang et al., 2023; Liang et al., 2023). Subsequent multimodal agents incorporated VLMs into robotic decision-making, enabling closed-loop task execution through visual feedback (Zhang et al., 2025a; Zhi et al., 2025). Agentic navigation has subsequently emerged as an important application, motivated in part by advances in visual grounding and spatial reasoning (Zhou et al., 2024; Chen et al., 2024a). Recent works have explored both zero-shot and trained agentic navigation: EvoNav (Dai et al., 2026) uses experience-based reasoning for zero-shot VLN-CE, while AgentVLN (Xin et al., 2026) develops a trained agentic VLN framework with a reusable skill library. Despite these advances, existing agentic navigation methods primarily target individual navigation tasks and do not provide a unified mechanism for coordinating heterogeneous objectives or reusing experience across tasks and environments. RING-Nav addresses this gap by introducing an RSI-based paradigm for training-free heterogeneous generalist navigation.

3 METHOD

3.1 FRAMEWORK OVERVIEW OF RECURSIVE SELF-IMPROVING HARNESS

We formulate generalist navigation as a training-free recursive self-improving (RSI) harness built upon frozen VLMs and fixed navigation tools. As illustrated in the left panel of Fig. 2, RING-Nav adopts a hierarchical architecture that separates *high-level task planning*, *real-time spatial perception*, and *planning-level tool orchestration*. Given a task specification g^τ, the high-level agentic task planner interprets the instruction and decomposes the overall goal into executable subtasks. In parallel, continuously running perception and localization modules process RGB-D observations to estimate the robot pose (Campos et al., 2021), detect objects (Cheng et al., 2024), and maintain geometric and semantic representations of the environment, including a 3D occupancy map (Jiang et al., 2026) and a semantic scene graph (Gu et al., 2024). The task planner and spatial perception modules jointly provide task progress, episodic memory, and multimodal spatial evidence, which are integrated into a *contextualized navigation state* (CNS) c_t^τ. Conditioned on the CNS, the agent dynamically orchestrates a shared set of planning-level navigation tools $\mathcal{U}$ across heterogeneous tasks. Meanwhile, an RSI experience graph $\mathcal{G}_{\text{exp}}^{(n)}$ accumulates cross-episode experience to guide subsequent tool decisions.

At each decision step, the harness constructs a set of feasible structured actions $\mathcal{A}_t$ from the current CNS c_t^τ. Each structured action $a_t = (u_t, \xi_t)$ specifies a navigation tool $u_t \in \mathcal{U}$ together with its execution parameters ξ_t. The frozen VLM evaluates these candidates based on the current task and CNS, while the experience graph independently retrieves historical records from navigation situations similar to the current one. The harness then combines current VLM reasoning with retrieved experience to select the next action:

$$a_t = (u_t, \xi_t) \in \mathcal{A}_t. \tag{1}$$

Executing a_t produces a new observation o_{t+1} and execution feedback, which are used to update the CNS from c_t^τ to c_{t+1}^τ. The agent can therefore revise its tool selection online as new spatial evidence, execution outcomes, and task progress become available.

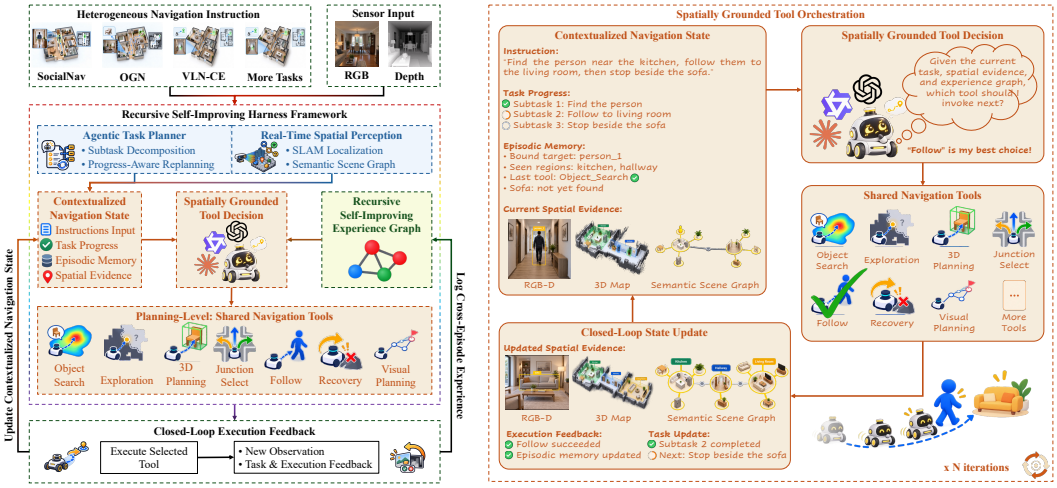


Figure 2: Overview of the RING-Nav framework. **Left:** The hierarchical harness integrates agentic task planning and real-time spatial perception into the CNS to orchestrate shared navigation tools. Execution feedback updates the CNS, while cross-episode experience accumulates in the RSI-EG to guide future tool decisions. **Right:** An example of a single tool-decision iteration. Each episode repeats this closed-loop process until all subtasks are completed.

The same interaction process drives cross-episode self-improvement. For each executed tool transition $u_i \to u_j$, the harness records the pre-transition navigation context and observed outcome in the corresponding conditional experience set $D_{ij}^{(n)}$. Upon episode completion, task-specific benchmark metrics are mapped to a common historical utility and associated with the executed transitions. These records are then incorporated into the experience graph, updating $\mathcal{G}_{\text{exp}}^{(n)}$ to $\mathcal{G}_{\text{exp}}^{(n+1)}$. Consequently, experience accumulated from previous episodes can be retrieved in subsequent episodes to improve tool orchestration across different tasks and environments.

The following sections describe the online tool-orchestration and cross-episode experience-evolution mechanisms in detail. Complete system architecture and tool implementation details are provided in the appendix A.

3.2 Spatially Grounded Tool Orchestration

Spatially grounded tool orchestration (SGTO) comprises four components, as illustrated in the right panel of Fig. 2:*contextualized navigation state*, which organizes the information required for each decision; *spatially grounded tool decision*, which selects the next tool and its execution arguments; *shared navigation tools*, which provide reusable capabilities across heterogeneous tasks; and *closed-loop state update*, which incorporates execution outcomes into the subsequent decision context.

Contextualized navigation state. The design of context is central to agentic navigation, as it determines what information is available when selecting the next action. To support heterogeneous tasks and dynamically changing environments, we introduce a navigation-specific *contextualized navigation state* (CNS) that unifies task progress, episodic experience, and multimodal spatial evidence within a shared decision context. This representation allows the agent to coordinate the same set of navigation tools as subtasks evolve and new spatial evidence becomes available.

At decision step t, we represent the CNS as $c_t^\tau = (q_t^\tau, \rho_t^\tau, \mathcal{M}_t^{\text{epi}}, \mathcal{E}_t^{\text{spa}})$, where q_t^τ denotes the active subtask, ρ_t^τ its current progress, $\mathcal{M}_t^{\text{epi}}$ the episodic memory, and $\mathcal{E}_t^{\text{spa}}$ the current spatial evidence. Specifically, $\mathcal{E}_t^{\text{spa}} = (\mathbf{I}_t, \mathbf{D}_t, \mathbf{T}_t, \mathcal{M}_t^{3D}, \mathcal{G}_t^{\text{sem}})$ consists of the current RGB image $\mathbf{I}_t$, depth observation $\mathbf{D}_t$, robot pose $\mathbf{T}_t$, 3D occupancy map $\mathcal{M}_t^{3D}$, and semantic scene graph $\mathcal{G}_t^{\text{sem}}$. Together, these components provide the information the agent uses for the current tool decision.

Spatially grounded tool decision. Given c_t^τ, the harness constructs the current feasible action set $\mathcal{A}_t = \{(u_j, \xi) \mid u_j \in \mathcal{U},\ \xi \in \Xi_j(c_t^\tau)\}$, where $\Xi_j(c_t^\tau)$ denotes the set of currently feasible arguments for tool u_j under the CNS c_t^τ. For example, it may contain currently trackable person identities for *Follow* or unexplored regions for *Exploration*. For each feasible candidate $(u_j, \xi) \in \mathcal{A}_t$, the frozen

VLM agent returns a relative execution likelihood $L_t(j, \xi) \in [0, 1]$ conditioned on the current task g^τ, CNS c_t^τ, feasible action set $\mathcal{A}_t$, and retrieved historical experience. A larger $L_t(j, \xi)$ indicates a stronger current preference for candidate (u_j, ξ). The final tool decision further combines this current reasoning with relevant experience from the RSI-EG, as detailed in Sec. 3.3.

Shared navigation tools. All tools in $\mathcal{U}$ operate at the planning level and are accessed through a unified interface, allowing the same tool set to be composed across heterogeneous navigation tasks. Our implementation includes *Object Search* for semantic target discovery (Gong et al., 2026; Li et al., 2023), *Exploration* for frontier-based exploration of unobserved space (Yamauchi, 1997), *3D Planning* for geometry-grounded goal reaching through occupancy-map path planning (Hart et al., 1968; Cheng et al., 2024), *Junction Select* for selecting among competing spatial branches using navigation experience, *Follow* for dynamic human following (Zhong et al., 2024), *Recovery* for returning to a previously visited safe position after execution failure, and *Visual Planning* for image-conditioned local navigation (Cai et al., 2026). Each tool receives structured arguments ξ and returns execution status together with updated observations. Implementation details are provided in the supplementary material.

Closed-loop state update. After executing the selected action $a_t = (u_t, \xi_t)$, the resulting observation o_{t+1} and execution feedback update task progress, episodic memory, and spatial evidence, yielding the next CNS c_{t+1}^τ. The agent then re-evaluates the active subtask and constructs a new feasible action set from the updated state. Thus, tool selection remains closed-loop: successful execution does not necessarily imply subtask completion, while newly acquired spatial evidence may alter the subsequent tool choice.

3.3 Recursive Self-Improving Experience Graph

While SGTO enables the agent to reason over the current navigation state, it does not by itself accumulate experience across episodes. We therefore introduce a *recursive self-improving experience graph* (RSI-EG), illustrated in Fig. 3, which augments current VLM reasoning with experience accumulated from previous tool transitions. The RSI-EG consists of four operations: *task-adaptive metric evaluation*, which converts heterogeneous episode outcomes into a common utility; *experience graph evolution*, which records tool transitions and their outcomes; *contextual experience retrieval*, which identifies historically similar navigation situations; and *experience-informed tool decision*, which integrates retrieved experience with the VLM's current preference.

Task-adaptive metric evaluation. Different navigation benchmarks expose different evaluation protocols. ObjectNav, for example, commonly reports success rate (SR) and success-weighted path length (SPL) (Ramakrishnan et al., 2021), while more comprehensive benchmarks additionally consider metrics such as navigation error (NE), normalized dynamic time warping (nDTW), human-following metrics (HFS/HFR), embodied question-answering accuracy (EQA-ACC), and semantic interaction index (SII) (Krantz et al., 2020; Sun et al., 2026). To accumulate experience across tasks with heterogeneous evaluation protocols, we map all available episode metrics into a common normalized utility space in which larger values indicate better outcomes.

Let $\mathcal{M}$ denote the global set of supported evaluation metrics. For episode k, let $r_{k,q}$ denote its result under metric $q \in \mathcal{M}$, $\delta_{k,q} \in \{0, 1\}$ indicate whether q is defined and available for the current task, and $\alpha_q \geq 0$ denotes a pre-specified metric weight. We define the task-adaptive historical utility as

$$G_k = \frac{\sum_{q \in \mathcal{M}} \delta_{k,q} \alpha_q \phi_q(r_{k,q})}{\sum_{q \in \mathcal{M}} \delta_{k,q} \alpha_q}, \tag{2}$$

where $\phi_q(\cdot)$ converts heterogeneous metrics to a common $[0, 1]$ higher-is-better scale. For bounded higher-is-better metrics, ϕ_q performs linear normalization. For lower-is-better metrics, we reverse their direction after normalization. We use $\phi_{\mathrm{SII}}(r) = 1 - r/100$ and $\phi_{\mathrm{NE}}(r) = \exp(-r/\sigma_{\mathrm{NE}})$, where σ_{NE} is a fixed navigation-error scale. Metrics that are not applicable to the current task are excluded by setting $\delta_{k,q} = 0$, while reported zero values remain valid observations.

All metric weights and normalization parameters are fixed before held-out evaluation. Importantly, benchmark ground truth is never exposed to the agent or used during online tool selection. The harness receives only the resulting scalar episode-level utility after execution. Consequently, neither the VLM nor the navigation tools are updated using task-specific ground-truth supervision, preserving

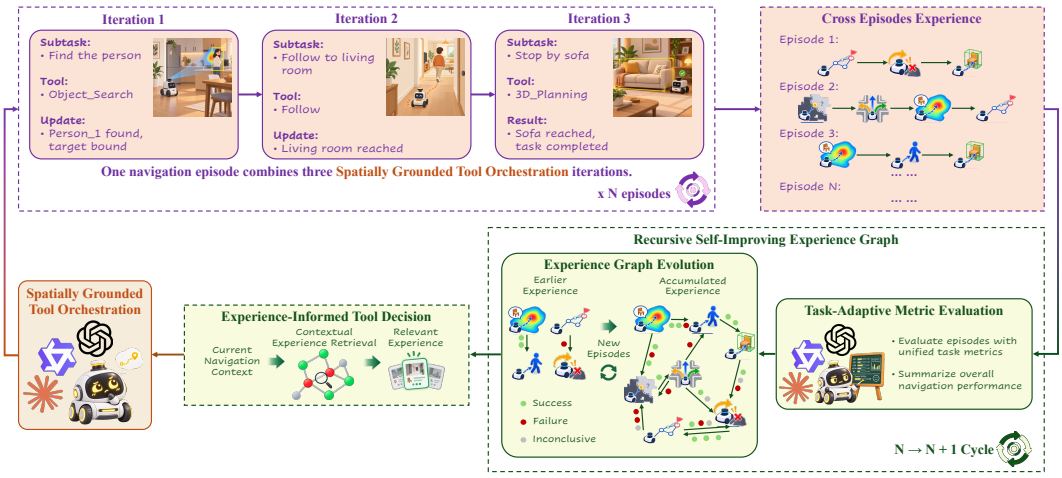


Figure 3: An example of RSI-EG guide tool orchestration. The **purple dashed box** illustrates multiple tool-decision iterations within a single episode, as detailed in Fig. 2. As episodes accumulate, the **green dashed box** shows how navigation outcomes are incorporated into the RSI-EG, which retrieves relevant experience to guide subsequent tool decisions.

the training-free nature of the framework. The resulting G_k serves solely as the historical utility associated with the experience generated by episode k.

Experience graph evolution. We represent the experience graph at evolution step n as $\mathcal{G}_{\text{exp}}^{(n)} = (\mathcal{U}, \mathcal{E}_{\text{exp}}^{(n)})$, where the nodes are the shared navigation tools in $\mathcal{U}$. Each directed edge $u_i \to u_j$ represents a tool transition that has been executed in previous episodes and is associated with a conditional experience set $D_{ij}^{(n)}$. An unobserved transition therefore has $D_{ij}^{(n)} = \varnothing$ but remains available to the online agent whenever its corresponding structured action belongs to $\mathcal{A}_t$. Given a candidate $(u_j, \xi) \in \mathcal{A}_t$, we construct a retrieval representation of the current CNS as $x_t^{j,\xi} = [\ell_t^{j,\xi}, v_t^{j,\xi}, g_t^{j,\xi}, h_t]$, where $\ell_t^{j,\xi}$ captures the current task condition, $v_t^{j,\xi}$ encodes the visual context, $g_t^{j,\xi}$ describes the candidate-conditioned local spatial geometry, and h_t summarizes recent execution behavior. Whenever the harness executes a transition $u_i \to u_j$, it records the pre-execution representation $x_t^{j,\xi}$ and the observed execution outcome. After the episode completes and is evaluated, its historical utility G_k is associated with the recorded transitions, which are then appended to the corresponding conditional experience sets $D_{ij}^{(n)}$. The accumulated records update the graph to $\mathcal{G}_{\text{exp}}^{(n+1)}$. Both successful and failed executions are retained as historical experience.

Contextual experience retrieval. For a candidate $(u_j, \xi) \in \mathcal{A}_t$, let u_i denote the most recently executed tool. We compare the current representation $x_t^{j,\xi}$ with historical records $x_r = [\ell_r, v_r, g_r, h_r]$ in $D_{ij}^{(n)}$ along the four components defined above. Specifically, we compute $d_T(\ell_t^{j,\xi}, \ell_r)$, $d_V(v_t^{j,\xi}, v_r)$, $d_G(g_t^{j,\xi}, g_r)$, and $d_H(h_t, h_r)$ for task, visual, geometric, and execution-history differences, respectively. Let d_f, $f \in \{T, V, G, H\}$, denote the corresponding component-wise distance. The overall distance is then

$$d\left(x_t^{j,\xi}, x_r\right) = \frac{\sum_{f\in\{T,V,G,H\}} \theta_f \eta_{t,r,f} d_f}{\sum_{f\in\{T,V,G,H\}} \theta_f \eta_{t,r,f}}, \tag{3}$$

where θ_f controls the contribution of each component and $\eta_{t,r,f} \in \{0, 1\}$ indicates whether the corresponding comparison is available. We exclude records without any valid common component from retrieval. The top-K most relevant historical records are retrieved according to the distance $d(x_t^{j,\xi}, x_r)$ from the current navigation situation:

$$\mathcal{N}_{ij}^{(n)}(x_t^{j,\xi}) = \text{TopK}_{r\in D_{ij}^{(n)}} \exp\left(-\frac{d(x_t^{j,\xi}, x_r)}{\tau}\right), \tag{4}$$

where τ controls the similarity decay.

Experience-informed tool decision. The retrieved records are aggregated into a context-conditioned historical value

$$Q_{ij}^{(n)}(x_t^{j,\xi}) = \frac{\kappa\mu_0 + \sum\limits_{r\in\mathcal{N}_{ij}^{(n)}(x_t^{j,\xi})} \exp\Big(-\frac{d(x_t^{j,\xi},x_r)}{\tau}\Big)\, G_{e(r)}}{\kappa + \sum\limits_{r\in\mathcal{N}_{ij}^{(n)}(x_t^{j,\xi})} \exp\Big(-\frac{d(x_t^{j,\xi},x_r)}{\tau}\Big)}, \tag{5}$$

where $e(r)$ denotes the episode containing record r, and $G_{e(r)}$ is its historical utility computed by Eq. 2. μ_0 is a neutral prior and κ controls its strength. We then estimate the effective historical support and its influence on the current decision:

$$\lambda_{ij}^{(n)}(x_t^{j,\xi}) = \lambda_{\max}\frac{\sum\limits_{r\in\mathcal{N}_{ij}^{(n)}(x_t^{j,\xi})} \exp\Big(-\frac{d(x_t^{j,\xi},x_r)}{\tau}\Big)}{\sum\limits_{r\in\mathcal{N}_{ij}^{(n)}(x_t^{j,\xi})} \exp\Big(-\frac{d(x_t^{j,\xi},x_r)}{\tau}\Big) + n_0}, \tag{6}$$

Here, $\lambda_{\max} \leq 1$ bounds the influence of historical experience and n_0 controls how quickly it increases with historical support. The current-task score $L_t(j,\xi)$ defined in Sec. 3.2 is updated after the VLM incorporates the retrieved experience. We further combine the updated VLM reasoning $L_t(j,\xi)$ with context-conditioned historical value $Q_{ij}^{(n)}(x_t^{j,\xi})$ to select the next structured action:

$$a_t = (u_t,\xi_t) = \arg\max_{(u_j,\xi)\in\mathcal{A}_t} \Big\{\Big[1-\lambda_{ij}^{(n)}(x_t^{j,\xi})\Big]\, L_t(j,\xi) + \lambda_{ij}^{(n)}(x_t^{j,\xi}) Q_{ij}^{(n)}(x_t^{j,\xi})\Big\}. \tag{7}$$

If no relevant historical record is available, $\lambda_{ij}^{(n)}(x_t^{j,\xi}) = 0$, so the decision relies on $L_t(j,\xi)$ based on VLM reasoning without retrieved experience. As new transition outcomes accumulate, the RSI-EG updates the historical guidance for future tool decisions while keeping the VLM model weights and navigation tools fixed.

4 Experiments

We evaluate RING-Nav across diverse navigation settings to assess its generality and self-improvement capability. Sec. 4.1 evaluates performance on individual navigation tasks, including instruction following and object navigation. Sec. 4.2 evaluates heterogeneous multi-stage navigation. Sec. 4.3 studies the effects of VLM backbones and key framework components. RSI-EG is populated using experience collected from development episodes and frozen before testing; no test-set interaction, benchmark metric, or evaluation-derived utility is written back into the graph. All VLM and navigation-module parameters remain frozen.

4.1 Evaluation on Individual Navigation Tasks

Performance on VLN-CE (Krantz et al., 2020). We evaluate instruction-following navigation on R2R-CE (Anderson et al., 2018) under the standard VLN-CE protocol used by recent zero-shot methods (Qiao et al., 2025). Table 1 compares RING-Nav with representative supervised and training-free approaches using success rate (SR), success weighted by path length (SPL), oracle success rate (OSR), normalized Dynamic Time Warping (nDTW), and navigation error (NE). RING-Nav-GPT5.5 achieves 62% SR and 52.47% SPL, improving over EvoNav (Dai et al., 2026) by 19.0 and 14.70 percentage points, respectively, while achieving the best SR, SPL, OSR, and nDTW among training-free methods. With the same GPT-4o backbone, RING-Nav improves SR from 30% to 60% and SPL from 24.91% to 52.20%, while reducing NE from 5.97 to 5.19. The consistent gains across VLM backbones highlight the contribution of the proposed navigation harness beyond the underlying VLM.

Performance on OGN (Batra et al., 2020). We further evaluate zero-shot object-goal navigation on HM3D (Yadav et al., 2023; Ramakrishnan et al., 2021) and MP3D (Chang et al., 2017) under the standard Habitat ObjectNav protocol. As shown in Table 2, RING-Nav achieves 67.0% SR and 33.8% SPL on HM3D, improving over ASCENT (Gong et al., 2026) by 1.6 and 0.3 percentage

Table 1: Performance comparison of the instruction-following navigation task on the R2R-CE benchmark. The best result in each metric is shown in **bold**.

Method	SR↑	SPL↑	OSR↑	nDTW↑	NE↓
Supervised Learning Methods					
CMA (CVPR'22) (Hong et al., 2022)	37	32.17	45	50.77	6.92
RecBERT (CVPR'22) (Hong et al., 2022)	48	43.22	57	54.81	5.80
BEVBert (ICCV'23) (An et al., 2023)	60	**53.41**	64	61.40	5.13
ETPNav (TPAMI'24) (An et al., 2024)	58	52.19	63	61.16	5.16
Training-Free Methods					
LXMERT (CVPR'22) (Hong et al., 2022)	2	1.87	22	18.73	10.84
MapGPT-CE-GPT4o (ACL'24) (Chen et al., 2024b)	7	5.04	21	–	8.16
DiscussNav-GPT4 (ICRA'24) (Long et al., 2024)	11	10.51	15	42.87	7.77
NavGPT-CE-GPT4 (AAAI'24) (Zhou et al., 2024)	16.3	10.20	26.9	–	8.37
Open-Nav-GPT4o (ICRA'25) (Qiao et al., 2025)	19	16.10	23	45.79	6.70
Open-Nav-Gemini-2.5-pro (ICRA'25) (Qiao et al., 2025)	23	19.90	30	49.51	7.28
EvoNav-GPT4o (CVPR'26) (Dai et al., 2026)	30	24.91	35	56.32	5.97
EvoNav-Gemini-2.5-pro (CVPR'26) (Dai et al., 2026)	43	37.77	51	62.38	**5.04**
RING-Nav-GPT4o	60	52.20	63	61.55	5.19
RING-Nav-GPT5.5	**62**	52.47	**66**	**64.84**	5.07

Table 2: Performance comparison on HM3D and MP3D ObjectNav. The best result in each metric is shown in **bold**.

	HM3D		MP3D	
Method	**SR↑**	**SPL↑**	**SR↑**	**SPL↑**
ZSON (NeurIPS'22) (Majumdar et al., 2022)	25.5	12.6	15.3	4.8
L3MVN (IROS'23) (Yu et al., 2023)	50.4	23.1	34.9	14.5
PixNav (ICRA'24) (Cai et al., 2024)	37.9	20.5	–	–
VLFM (ICRA'24) (Yokoyama et al., 2024)	50.9	23.6	32.5	15.9
SG-Nav (NeurIPS'24) (Yin et al., 2024)	54.0	24.9	40.2	16.0
ActPept (RAL'24) (Guo et al., 2024)	–	–	39.8	17.4
InstructNav (CoRL'24) (Long et al., 2025)	58.0	20.9	–	–
ApexNav (RAL'25) (Zhang et al., 2025e)	59.6	33.0	39.2	**17.8**
MFNP (ICRA'25) (Zhang et al., 2025d)	58.3	26.7	41.1	15.4
ASCENT (RAL'26) (Gong et al., 2026)	65.4	33.5	44.5	15.5
RING-Nav	**67.0**	**33.8**	**48.5**	17.5

points. On MP3D, it achieves 48.5% SR and 17.5% SPL, exceeding ASCENT by 4.0 and 2.0 percentage points. Compared with ApexNav (Zhang et al., 2025e), which obtains higher SPL on MP3D, RING-Nav improves SR by 9.3 percentage points while maintaining comparable path efficiency.

ObjectNav is particularly challenging for agentic navigation because success depends heavily on spatial perception and exploration, while the sparse goal specification provides limited scope for language-based reasoning. RING-Nav addresses this through tool-level spatial decision-making: *ObjectSearch* combines semantic and geometric cues for frontier exploration, while *JunctionSelect* leverages context-matched experience to avoid ineffective branches and redundant exploration. Despite being designed for generalist navigation, RING-Nav therefore transfers effectively to task-specific object navigation.

4.2 Evaluation on Heterogeneous Navigation Tasks

We evaluate heterogeneous generalist navigation on the Unitree AlienGo subset of OmniNavBench (Sun et al., 2026) in MP3D environments. Unlike individual navigation tasks, each instruction combines language-guided navigation, object search, relative-distance movement, human following, and person avoidance, requiring coordinated tool execution and continuous subtask tracking. As shown in Table 3, RING-Nav achieves 28.75% SR and 21.24% SPL, improving over OmniNav (Xue et al., 2026) by 5.0 and 0.83 percentage points. More importantly, subgoal comple-

Table 3: Performance comparison on the OmniNavBench MP3D–AlienGo subset. Higher is better for all metrics except SII. The best result in each metric is shown in **bold**.

Method	SR↑	SPL↑	OSR↑	CSR↑	SGC↑	HFS↑	HFR↑	SII↓
MTU3D (ICCV'25) (Zhu et al., 2025)	0.00	0.00	10.00	2.50	28.17	0.00	14.19	8.52
PoliFormer (CoRL'24) (Zeng et al., 2025)	0.00	0.00	17.50	2.50	24.27	13.95	9.80	**4.11**
Uni-NaVid (RSS'25) (Zhang et al., 2025c)	3.75	1.85	10.00	0.00	29.45	4.65	22.66	13.70
OmniNav (ICLR'26) (Xue et al., 2026)	23.75	20.41	28.75	10.00	43.91	16.28	28.77	19.08
RING-Nav	**28.75**	**21.24**	**33.75**	**13.75**	**57.33**	**53.48**	**69.65**	9.83

Table 4: Ablation studies on VLM backbones and framework components across four navigation benchmarks. The best result in each metric is shown in **bold**.

Method	R2R-CE		OGN-HM3D		OGN-MP3D		OmniNavBench	
	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑	SR↑	SPL↑
RING-Nav-GPT5.5	**62.0**	**52.47**	**67.0**	**33.8**	**48.5**	**17.5**	**28.75**	**21.24**
VLM Backbone Ablation								
RING-Nav-GPT4o	60.0	52.20	66.7	32.2	47.9	17.1	**28.75**	19.26
RING-Nav-Qwen2.5-VL-7B	57.0	47.98	66.5	32.4	48.1	16.8	27.50	17.04
Component Ablation								
RING-Nav (w/o CNS, RSI-EG)	24.0	12.70	49.2	19.9	33.5	10.2	–	–
RING-Nav (w/o RSI-EG)	37.0	35.60	53.3	26.0	34.9	13.3	12.50	8.41

tion (SGC) increases from 43.91% to 57.33%, while human-follow success (HFS) and human-follow ratio (HFR) improve by 37.20 and 40.88 percentage points, respectively. These gains demonstrate the ability of RING-Nav to coordinate heterogeneous tools and sustain effective decision-making over long-horizon, multi-stage tasks.

4.3 Ablation Study

VLM Backbone Ablation. Table 4 shows that RING-Nav remains effective across VLM backbones. Replacing GPT-5.5 with Qwen2.5-VL-7B reduces SR by only 5.0 percentage points on R2R-CE and less than 1 percentage point on both HM3D and MP3D ObjectNav. On OmniNavBench, GPT-4o and Qwen2.5-VL-7B achieve 28.75% and 27.50% SR, respectively. These results indicate that RING-Nav is not tightly coupled to a particular VLM backbone.

Component Ablation. Removing CNS and RSI-EG substantially degrades performance, while adding each component yields consistent gains. Introducing CNS increases SR from 24.0% to 37.0% on R2R-CE and from 49.2% to 53.3% on HM3D. Adding RSI-EG further increases SR to 62.0%, 67.0%, and 48.5% on R2R-CE, HM3D, and MP3D, respectively. On OmniNavBench, RSI-EG more than doubles SR from 12.50% to 28.75% and increases SPL from 8.41% to 21.24%. These results demonstrate the complementary roles of spatially grounded context and cross-episode experience in effective tool orchestration.

5 Discussion and Conclusion

We presented RING-Nav, a recursive self-improving (RSI) harness for training-free heterogeneous generalist navigation. By integrating task planning and real-time spatial perception into a contextualized navigation state (CNS), RING-Nav enables a unified agent to coordinate shared navigation tools across diverse objectives and environments. Combined with the RSI experience graph (RSI-EG), spatially grounded tool orchestration (SGTO) allows the agent to reuse cross-episode experience and adapt its decisions without updating model parameters. Experiments on VLN-CE, ObjectNav, and OmniNavBench demonstrate strong performance across both individual and heterogeneous multi-stage navigation tasks. Future work will investigate more adaptive experience-graph management, including experience selection, retention, and pruning, together with more efficient retrieval mechanisms for scalable long-term self-improvement.

AI USE STATEMENT

We used generative AI tools to assist with English-language editing, including improving the manuscript's grammar, clarity, and readability. The tools were not used to formulate research hypotheses, design the research methodology or experiments, or interpret the experimental results All AI-assisted revisions were reviewed and verified by the authors. The authors take full responsibility for the content of this paper.

REFERENCES

Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. BEVBert: Multimodal map pre-training for language-guided navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2737–2748, 2023.

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.

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, June 2018.

Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *CoRR*, abs/2006.13171, 2020.

Wenzhe Cai, Siyuan Huang, Guangran Cheng, Yuxing Long, Peng Gao, Changyin Sun, and Hao Dong. Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill. In *2024 IEEE International Conference on Robotics and Automation*, pp. 5228–5234, 2024.

Wenzhe Cai, Jiaqi Peng, Yuqiang Yang, Yujian Zhang, Meng Wei, Hanqing Wang, Yilun Chen, Tai Wang, and Jiangmiao Pang. Navdp: Learning sim-to-real navigation diffusion policy with privileged information guidance. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. doi: 10.1109/TRO.2021.3075644.

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. In *2017 International Conference on 3D Vision (3DV)*, pp. 667–676, 2017.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14455–14465, June 2024a.

Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee Wong. MapGPT: Map-guided prompting with adaptive path planning for vision-and-language navigation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 9796–9810, 2024b.

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, June 2022.

Ziyi Chen, Yingnan Guo, Zedong Chu, Minghua Luo, Yanfen Shen, Mingchao Sun, Junjun Hu, Shichao Xie, Yang Kuan, Pei Shi, Zhining Gu, Lu Liu, Honglin Han, Xiaolong Wu, Mu Xu, and Yu Zhang. SocialNav: Training human-inspired foundation model for socially-aware embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28796–28806, June 2026.

Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16901–16911, 2024. doi: 10.1109/CVPR52733.2024.01599.

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 (CVPR)*, pp. 15177–15187, June 2026.

Chen Gao, Liankai Jin, Xingyu Peng, Jiazhao Zhang, Yue Deng, Annan Li, He Wang, and Si Liu. Octonav: Towards generalist embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 40074–40084, June 2026a.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Qihan Ren, Yiran Wu, Hongru Wang, Han Xiao, Yuhang Zhou, Shaokun Zhang, Jiayi Zhang, Jinyu Xiang, Yixiong Fang, Qiwen Zhao, Dongrui Liu, Cheng Qian, Zhenhailong Wang, Minda Hu, Huazheng Wang, Qingyun Wu, Heng Ji, and Mengdi Wang. A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence. *Transactions on Machine Learning Research*, 2026b.

Peng Gao, Peng Wang, Feng Gao, Fei Wang, and Ruyue Yuan. Vision-language navigation with embodied intelligence: A survey. *TMLR*, 2024.

Zeying Gong, Rong Li, Tianshuai Hu, Ronghe Qiu, Lingdong Kong, Lingfeng Zhang, Guoyang Zhao, Yiyi Ding, and Junwei Liang. Stairway to success: An online floor-aware zero-shot object-goal navigation framework via llm-driven coarse-to-fine exploration. *IEEE Robotics and Automation Letters*, 11(3):2943–2950, 2026.

Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5021–5028, 2024.

Yu Guo, Jinsheng Sun, Ruiheng Zhang, Zhiqi Jiang, Zhenqiang Mi, Chao Yao, Xiaojuan Ban, and Mohammad S. Obaidat. An object-driven navigation strategy based on active perception and semantic association. *IEEE Robotics and Automation Letters*, 9(8):7110–7117, 2024.

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. 13137–13146, June 2020.

Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. doi: 10.1109/TSSC.1968.300136.

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*, pp. 15418–15428, 2022. doi: 10.1109/CVPR52688.2022.01500.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning

through planning with language models. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1769–1782. PMLR, 2023.

Brian Ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander T. Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Nikhil J. Joshi, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do as i can, not as i say: Grounding language in robotic affordances. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 287–318. PMLR, 2023.

Zeyu Jiang, Changqing Zhou, Xingxing Zuo, and Changhao Chen. Freeocc: Training-free embodied open-vocabulary occupancy prediction. In *Robotics: Science and Systems (RSS)*, 2026.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the navgraph: Vision-and-language navigation in continuous environments. In *Computer Vision – ECCV 2020*, volume 12373 of *Lecture Notes in Computer Science*, pp. 104–120. Springer, 2020. doi: 10.1007/978-3-030-58604-1_7.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742. PMLR, 2023.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500. IEEE, 2023.

Jiaqi Liu, Kaiwen Xiong, Peng Xia, Yiyang Zhou, Haonian Ji, Lu Feng, Siwei Han, Mingyu Ding, and Huaxiu Yao. Agent0-vl: Exploring self-evolving agent for tool-integrated vision-language reasoning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Yuxing Long, Xiaoqi Li, Wenzhe Cai, and Hao Dong. Discuss before moving: Visual language navigation via multi-expert discussions. In *2024 IEEE International Conference on Robotics and Automation*, pp. 17380–17387, 2024.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2049–2060. PMLR, 06–09 Nov 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. In *Advances in Neural Information Processing Systems*, volume 35, pp. 32340–32352, 2022.

Neel Mokaria, Rishie Raj, Dheeraj Baiju, Xiaoqian Shen, Shraman Pramanick, Kevin Qinghong Lin, Arda Senocak, Mike Zheng Shou, Philip Torr, Mohamed Elhoseiny, Yapeng Tian, Ruohan Gao, Salman Khan, Sayan Nag, Sanjoy Chowdhury, and Dinesh Manocha. A survey on foundations and frontiers of multimodal agentic frameworks: Techniques and applications. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856.

Michael Joseph Munje, Chen Tang, Shuijing Liu, Zichao Hu, Yifeng Zhu, Jiaxun Cui, Garrett Warnell, Joydeep Biswas, and Peter Stone. SocialNav-SUB: Benchmarking vlms for scene understanding in social robot navigation. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 1120–1143. PMLR, 27–30 Sep 2025.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

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. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6710–6717. IEEE, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

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.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of EMNLP-IJCNLP*, pp. 3982–3992, 2019.

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. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 16923–16930. IEEE, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Daeun Song, Jing Liang, Amirreza Payandeh, Amir Hossain Raj, Xuesu Xiao, and Dinesh Manocha. VLM-Social-Nav: Socially aware robot navigation through scoring using vision-language models. *IEEE Robotics and Automation Letters*, 10(1):508–515, 2025.

Samson Sun, Tianyi Yang, Tengyue Wang, Yikai Xue, Zhengjie Xu, Lingming Zhang, Qichen Zhang, Chao Liang, and Zhipeng Zhang. Beyond isolation: A unified benchmark for general-purpose navigation. *arXiv preprint arXiv:2605.09441*, 2026.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Liuyi Wang, Kai Sheng, Zongtao He, Jinlong Li, Yongrui Qin, Haojie Dai, Xiangyi Wang, Jingwei Yang, Qingqing Yan, Chengju Liu, et al. A comprehensive survey and systematic real-world evaluation of embodied vision-and-language navigation. *IEEE Transactions on Automation Science and Engineering*, 2026.

Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Shawn Ma, and Yitao Liang. Describe, explain, plan and select: interactive planning with llms enables open-world multi-task agents. *Advances in Neural Information Processing Systems*, 36:34153–34189, 2023.

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, Xihui Liu, and Jiangmiao Pang. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

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. *arXiv preprint arXiv:2603.17670*, 2026.

Chaoran Xiong, Yulong Huang, Fangwen Yu, Changhao Chen, Yue Wang, Songpengchen Xia, and Ling Pei. Sensing, social, and motion intelligence in embodied navigation: A comprehensive survey. *arXiv preprint arXiv:2508.15354*, 2025.

Xinda Xue, Junjun Hu, Minghua Luo, Shichao Xie, Jintao Chen, Zixun Xie, Quan Kuichen, Wei Guo, Zedong Chu, Mu Xu, et al. Omninav: A unified framework for prospective exploration and visual-language navigation. In *International Conference on Learning Representations*, volume 2026, pp. 106216–106231, 2026.

Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, Alexander William Clegg, and Devendra Singh Chaplot. Habitat-matterport 3d semantics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4927–4936, 2023.

Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 146–151. IEEE, 1997. doi: 10.1109/CIRA.1997.613851.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 5285–5307, 2024.

Hang Yin, Xiuwei Xu, Linqing Zhao, Ziwei Wang, Jie Zhou, and Jiwen Lu. Unigoal: Towards universal zero-shot goal-oriented navigation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19057–19066. IEEE, 2025a.

Xunjian Yin, Xinyi Wang, Liangming Pan, Li Lin, Xiaojun Wan, and William Yang Wang. Gödel agent: A self-referential agent framework for recursively self-improvement. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27890–27913, Vienna, Austria, 2025b. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.1354.

Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3MVN: Leveraging large language models for visual target navigation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3554–3560, 2023.

Kuo-Hao Zeng, Zichen Zhang, Kiana Ehsani, Rose Hendrix, Jordi Salvador, Alvaro Herrasti, Ross Girshick, Aniruddha Kembhavi, and Luca Weihs. PoliFormer: Scaling on-policy rl with transformers results in masterful navigators. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 408–432. PMLR, 2025.

Jiaqi Zhang, Zinan Wang, Jiaxin Lai, and Hongfei Wang. Gptarm: An autonomous task planning manipulator grasping system based on vision–language models. *Machines*, 13(3):247, 2025a.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. Aflow: Automating agentic workflow generation. In *International Conference on Learning Representations*, volume 2025, pp. 34040–34077, 2025b.

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. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.079.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *Robotics: Science and Systems*, 2025c.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, et al. Embodied navigation foundation model. In *International Conference on Learning Representations*, volume 2026, pp. 127293–127322, 2026.

Lingfeng Zhang, Hao Wang, Erjia Xiao, Xinyao Zhang, Qiang Zhang, Zixuan Jiang, and Renjing Xu. Multi-floor zero-shot object navigation policy. In *2025 IEEE International Conference on Robotics and Automation*, pp. 6416–6422, 2025d.

Mingjie Zhang, Yuheng Du, Chengkai Wu, Jinni Zhou, Zhenchao Qi, Jun Ma, and Boyu Zhou. Apexnav: An adaptive exploration strategy for zero-shot object navigation with target-centric semantic fusion. *IEEE Robotics and Automation Letters*, 10(11):11530–11537, 2025e.

Peiyuan Zhi, Zhiyuan Zhang, Yu Zhao, Muzhi Han, Zeyu Zhang, Zhitian Li, Ziyuan Jiao, Baoxiong Jia, and Siyuan Huang. Closed-loop open-vocabulary mobile manipulation with gpt-4v. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4761–4767. IEEE, 2025.

Fangwei Zhong, Kui Wu, Hai Ci, Churan Wang, and Hao Chen. Empowering embodied visual tracking with visual foundation models and offline rl. In *European Conference on Computer Vision*, pp. 139–155. Springer, 2024. doi: 10.1007/978-3-031-73464-9_9.

Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(7):7641–7649, 2024.

Ziyu Zhu, Xilin Wang, Yixuan Li, Zhuofan Zhang, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Wei Liang, Qian Yu, Zhidong Deng, et al. Move to understand a 3d scene: Bridging visual grounding and exploration for efficient and versatile embodied navigation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8120–8132. IEEE, 2025.

A Appendix

A.1 System Architecture and Implementation

RING-Nav is implemented on a modular runtime, where perception, mapping, navigation, and robot execution are organized as independently scheduled modules through a Blueprint. The navigation tools are registered as Skills and exposed to the Agent through the Model Context Protocol (MCP). The MCP server provides the tool names, descriptions, and argument schemas, allowing the Agent to invoke heterogeneous navigation modules through a unified interface. When it receives an MCP tool call, the corresponding Skill validates its arguments and communicates with the underlying modules through their RPC interfaces. MCP serves as the agent-facing tool interface, while RPC is used internally by each Skill to invoke the underlying modules Continuous sensor observations, spatial representations, and control messages are exchanged through typed input and output streams rather than being routed through MCP. This separates high-level tool invocation from the robot runtime's internal data exchange and execution mechanisms.

The modules execute asynchronously at their respective operating frequencies. Sensor-driven perception and mapping modules update the spatial state as observations arrive, while the motion executor continues tracking the latest accepted trajectory independently of VLM inference. The Agent is invoked only when a higher-level decision is required, such as tool termination, target loss, blocked motion, or subtask transition. At each decision event, the Agent-facing runtime assembles a context snapshot from the current task state, available spatial evidence, and recent execution feedback.

The Agent then selects a Skill and issues a structured MCP request without directly controlling the sensor-processing or motion-execution loops.

MCP tool invocation is decoupled from the robot control loop: an outstanding Agent request does not require perception and control modules to suspend their ongoing operations. When the runtime receives a tool request, it checks the task identifier, arguments, and referenced spatial state before dispatching it to the corresponding module. The motion executor continuously tracks the accepted trajectory while perception and mapping operate asynchronously. Local replanning is triggered only when the current trajectory becomes infeasible, or execution is interrupted. The runtime reports the accepted tool request, actual execution status, and resulting observations separately, providing the feedback required for subsequent Agent decisions and experience logging.

A.2 Navigation Tool Implementation

The seven navigation tools are implemented as MCP-accessible Skills. Each Skill receives structured arguments from the Agent, queries the required runtime state, invokes the corresponding navigation module, and returns its execution status. Motion-producing Skills share a common robot execution interface for sending trajectories or velocity commands to the controller.

Exploration. Exploration performs goal-agnostic free-space exploration to actively reveal unknown regions and incrementally expand the spatial map. Following classical frontier-based exploration (Yamauchi, 1997), it identifies boundaries between observed free space and unexplored space, groups them into candidate exploration regions, and selects reachable observation points. The selected point is passed to the geometric planner for execution, while newly acquired observations are continuously fused into the map and update the frontier set. This exploration–mapping loop continues until sufficient free space is revealed or another navigation capability is selected.

Object Search. Object Search performs target-directed exploration for locating a specified object. Given a target description, BLIP-2 image–text matching (Li et al., 2023) estimates the semantic relevance between the current RGB observation and the target. The resulting score is fused with RGB-D observations to maintain a target-conditioned spatial value map, so unexplored regions can be prioritized by their likelihood of containing the target. Rather than selecting frontiers purely for map expansion, Object Search biases the exploration candidates produced by the Exploration module toward regions with higher accumulated target relevance, while also considering observable unknown area, path cost, and within-episode visitation and failure records, following recent value-guided zero-shot ObjectNav strategies (Gong et al., 2026). When multiple exploration branches remain plausible, Junction Select retrieves cross-episode experience from RSI-EG using the current visual appearance, local 3D geometry, task context, and recent execution history, and constrains Object Search to the corresponding subset of candidate regions. The shared navigation planner then executes the final target-conditioned exploration goal.

3D Planning. Our framework maintains an online scene graph together with a 3D spatial map, following the object-centric spatial representations commonly used in 3D scene-graph systems (Cheng et al., 2024; Gu et al., 2024). Previously observed objects are associated with persistent 3D locations, enabling direct navigation to already localized targets. Given the robot position $\mathbf{p}_t$ and a mapped target position $\mathbf{p}_g$, 3D Planning performs point-to-point A* search (Hart et al., 1968) over the traversable map to obtain a collision-free path between the two 3D locations. The resulting path is then passed to the local controller for execution.

Visual Planning. Visual Planning is used when the target object has already appeared in the current camera view but has not yet been reliably localized in the 3D scene graph or spatial map. Instead of waiting for a persistent 3D target position, the detected target is represented as a PixelGoal (u_t, v_t) in the image. Together with the current RGB-D observation, the PixelGoal is converted into an egocentric local goal and passed to NavDP (Cai et al., 2026), which generates a short-horizon trajectory for approaching the visually observed target while avoiding nearby obstacles. The resulting local trajectory is passed to the underlying motion controller for execution, providing a direct visual-to-motion pathway that can either drive the robot directly to the target or transition to 3D Planning once a reliable 3D target estimate becomes available.

Follow. Follow handles interactive navigation with a moving person through closed-loop visual tracking and reactive motion control. Given the person specified by the current instruction, the sys-

tem first establishes and maintains the target identity across successive observations using visual tracking (Zhong et al., 2024). Follow directly converts the tracked person state into linear and angular velocity commands. In the default image-space branch, the tracked bounding box's horizontal position and scale provide the control signals. The displacement of the bounding-box center from the image center determines the turning direction and angular velocity, while the apparent target scale regulates forward velocity and maintains an appropriate following distance. When reliable RGB-D geometry is available, the tracked region can also be back-projected into 3D to estimate the person's relative position. The resulting distance and bearing provide a geometric alternative to the image-space cues, but are likewise converted directly into forward and angular velocities. When the geometric estimate is unavailable, the controller falls back to the lightweight image-space following branch. In OmniNavBench (Sun et al., 2026), the resulting motion commands are executed through the simulator's native AlienGo locomotion interface. The benchmark provides a pretrained low-level locomotion controller that accepts commanded forward, lateral, and angular velocities and converts them into quadruped joint actions for physics-based execution. Our interface therefore only adapts the short-horizon motion commands produced by Follow into the benchmark-compatible control format and duration, without requiring an additional learned locomotion policy.

Junction Select. Junction Select is a non-motion Skill invoked when the scene graph (Gu et al., 2024) exposes multiple reachable exploration branches. For each candidate branch, the Skill extracts branch-specific visual and local geometric representations from the current observation and spatial map, then independently matches them against k previously executed branch records stored in RSI-EG. Each current–historical branch pair is evaluated according to task semantics, visual appearance, local geometry, and recent execution history, producing a conditional similarity weight that associates the historical outcome G_k with the current branch. The weighted historical outcomes are then aggregated into a branch-specific experience value, and the branch with the strongest supported preference is returned to the Agent together with its scene-graph spatial reference. Junction Select does not generate motion; the selected branch is then used to constrain downstream exploration and navigation Skills.

Recovery. Recovery is triggered when the Agent identifies an execution failure, such as blocked motion or loss of navigation progress. The runtime retrieves the most recent previously valid robot state and generates a backtracking motion toward that position. After returning to a stable state, control returns to the Agent for a new tool decision rather than continuing the failed plan.

Task decomposition and progress tracking

Input: [full instruction].
Decompose the instruction into ordered, independently verifiable subgoals. Preserve target attributes, spatial and temporal constraints, dependencies, and explicit stopping requirements. Do not convert navigation actions or tool calls into subgoals.
For each subgoal, maintain an online task record containing its target, constraints, completion requirements, current status, supporting evidence, and dependency on preceding subgoals. A subgoal may be marked complete only when its completion requirements are supported by current valid evidence.

```
{
  "task_id": "task-2",
  "instruction": "approach the chair in the hall and stop",
  "target": {"category": "chair", "region": "hall"},
  "depends_on": ["task-1"],
  "status": "active",
  "completion_requirements": [
      "target_confirmed",
      "approach_verified",
      "stop_required"
  ],
  "evidence_refs": ["obj-17", "obs-153"],
  "unmet_requirements": ["approach_verified"],
  "parent_task_id": null,
  "task_epoch": 2
}
```

A.3 Task State and Episodic Memory

Task decomposition and online task state. At the beginning of an episode, the VLM decomposes the full instruction into a sequence of independently verifiable subgoals. The decomposition specifies *what* must be achieved rather than prescribing a fixed tool sequence: spatial relations, target attributes, ordering constraints, and stopping requirements are retained within the corresponding subgoal, while low-level turns or movements are not introduced as separate tasks. Consequently, different Skills may pursue the same subgoal as observations and spatial knowledge evolve.

After each Skill execution, the resulting observation and execution feedback update the task state rather than directly advancing the task index. The VLM proposes whether to continue, verify, recover, or mark the current subgoal as a completion candidate; the task manager advances only after the referenced evidence remains valid and satisfies the corresponding completion requirements.

Episodic memory construction. RSI-EG stores conditional experience over the shared Skill graph rather than complete navigation trajectories or reusable routes. Following the notation in the main text, the experience graph at evolution step n is $\mathcal{G}_{\text{exp}}^{(n)} = (\mathcal{U}, \mathcal{E}_{\text{exp}}^{(n)})$, and each directed transition $u_i \to u_j$ is associated with an experience set $D_{ij}^{(n)}$. For an actually executed candidate (u_j, ξ) following Skill u_i, we record the pre-execution CNS representation $x_t^{j,\xi} = [\ell_t^{j,\xi}, v_t^{j,\xi}, g_t^{j,\xi}, h_t]$. The four components come from the online CNS and are serialized in modality-specific forms.

For the task component $\ell_t^{j,\xi}$, we encode the full instruction, current subgoal, and candidate goal using a frozen lightweight `all-MiniLM-L6-v2` sentence encoder (Reimers & Gurevych, 2019), producing a 384-D semantic embedding. We retain the current navigation phase and candidate execution mode as discrete attributes, so the task representation captures both semantic intent and execution context. Throughout RSI-EG, we compute all text-based semantic similarities using the same frozen encoder and cosine similarity, ensuring consistent representations across task conditions and subsequent textual comparisons. Thus, task memory stores the semantic embedding together with its phase, mode, and validity metadata, rather than the raw task identifier.

For the visual component $v_t^{j,\xi}$, the current front-view image and, when available, the task-relevant target crop are encoded by the frozen CLIP-B/32 (Radford et al., 2021) encoder into two L2-normalized 512-D features. The resulting scene–target representation is stored as a 2×512 feature array, along with its encoder version and per-role validity; by default, we do not retain the raw historical RGB image.

For the geometric component $g_t^{j,\xi}$, we first transform the current spatial memory M_t into the robot-centric frame and crop the observed same-floor region within a radius of 3 m. The candidate-conditioned extractor ϕ_G then converts this heterogeneous spatial state into a fixed local geometry descriptor, $g_t^{j,\xi} = [c_f, c_l, c_r, w, o, u, \Delta r, \Delta f, \Delta z, q_d, a_b, a_m, \rho]$, where c_f, c_l, c_r are the known traversable distances in the front, left, and right directions; w is the minimum observed passage width along the candidate path; o and u are the occupied and unknown fractions of the local map; $(\Delta r, \Delta f, \Delta z)$ gives the candidate anchor position in the robot-centric right–forward–up frame; q_d is the valid-depth fraction of the target bounding box; a_b and a_m denote the age of the target geometry and the supporting map observations; and ρ denotes whether the candidate is currently known to be reachable or blocked. The first twelve fields are stored as `float32` values, while reachability is stored as a categorical `int8` field. Each field is stored with validity and observation-coverage metadata, while the relative-position fields additionally retain an `anchor_kind` specifying whether the anchor is an observed object, observation point, or tracked person. The complete occupancy map, Scene Graph, and raw point cloud remain in online spatial memory; RSI-EG stores only the resulting candidate-conditioned descriptor and its associated metadata for cross-episode retrieval.

For the execution-history component h_t, the system summarizes recent measured robot behavior rather than storing an unrestricted action log. Before each decision, it aggregates a fixed recent window of robot motion, along with the most recent executed Skills, into six groups: net displacement, traveled distance, continuous stationary-motion duration, region re-entry count, target-person tracking state, and the ordered sequence of recent Skill executions and termination types. The resulting structured history summary, together with its validity and temporal coverage, is stored as the historical representation used by d_H. This representation allows RSI-EG to distinguish visually and

geometrically similar situations with different recent execution behaviors, such as normal progress, repeated exploration, or stalled motion.

Each executed transition additionally records the selected Skill and parameters, measured execution feedback, and the resulting post-execution state. After offline evaluation, the corresponding utility $G_{e(r)}$ is attached to the executed record and incorporated into the appropriate conditional experience set $D_{ij}^{(n)}$. We store only executed transitions, including both successful and failed cases. As in the main text, the retrieval representation excludes scene and episode identities, absolute coordinates, GT routes, and hidden answers.

pisodic transition memory

Schematic record for an executed Skill transition:

```
{
  "source_skill": "object_search",
  "executed_skill": "3d_planning",
  "parameters": {"target_ref": "obj_17"},

  "condition": {
    "task": {
      "feature": F_T(U, u_t, GoalText(xi)),
      "phase": "approach",
      "mode": "target_navigation"
    },

    "visual": {
      "feature": "float32[2,512]",
      "roles": ["scene", "target"],
      "valid": [true, true]
    },

    "geometry": {
      "descriptor": "local_geometry_context_v1",
      "fields": "12 float + 1 categorical",
      "valid": "per-field"
    },

    "history": {
      "summary":
        [displacement, path_length, stationary_time,
         reentry, person_state, recent_skills],
      "valid": "per-field"
    }
  },

  "execution": {
    "outcome": "completed",
    "duration_s": 4.2,
    "path_length_m": 1.6
  },

  "utility": G_e(r)
}
```