RSI-NAV: RECURSIVE SELF-IMPROVEMENT WITH PROACTIVE PREFLECTION FOR TRAINING-FREE OBJECT-GOAL NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Training-Free Object-Goal Navigation (TF-OGN) requires embodied agents to explore and locate target objects in unseen environments without any task-specific training. To this end, recent methods leverage foundation models. However, many of the compared TF-OGN methods do not explicitly maintain memory conditioned on outcomes across episodes, limiting systematic reuse of completed experience. In this paper, we propose RSI-Nav, a TF-OGN framework that performs recursive self-improvement (RSI) by using completed navigation outcomes to update the rule memory guiding future decisions. Specifically, we build an agentic rule memory by extracting actionable knowledge from past trajectories. We retrieve rules by combining trajectory-derived semantic credit and historical success through UCB-based utility and exploration. In addition, we introduce a memory-guided preflection module that forecasts potential outcomes before action, reducing inefficient exploration. Extensive experiments show that our method outperforms existing training-free baselines, achieving a 10.1% relative improvement in success rate on MP3D.

1 INTRODUCTION

Object-Goal Navigation (OGN) is a fundamental task in embodied artificial intelligence (Savva et al., 2019; Sun et al., 2024). It requires an agent to perceive, reason, and act within an unseen space and locate a specific target. Existing OGN methodologies fall into training-based and training-free paradigms. Training-based methods (Chen et al., 2022; Ramrakhya et al., 2023; Dang et al., 2023; Wijmans et al., 2019; Wasserman et al., 2024; Tan et al., 2024) rely on Reinforcement Learning (RL) or Imitation Learning (IL) and demand massive datasets or extensive simulation. This leads to sample inefficiency and poor generalization in unseen environments. To address this, recent training-free methods (Zhang et al., 2025; Gong et al., 2026; Cai et al., 2024; Yang et al., 2026) leverage foundation models like Large Language Models and Vision-Language Models (LLMs/VLMs) as semantic planners. However, many of the compared TF-OGN methods rely on fixed semantic priors and do not explicitly maintain memory conditioned on outcomes across episodes, which can lead to repeated mistakes. Another key limitation is the high cost of trial and error. Under limited step budgets, each action is expensive. Existing methods often rely on post-hoc correction (Kim et al., 2024; Wang et al., 2024a), where errors are detected only after execution, leading to wasted steps. Efficient agents should therefore anticipate failures before execution.

These limitations motivate a framework that turns past navigation outcomes into guidance for future decisions. Specifically, we propose a training-free OGN framework, RSI-Nav, that performs recursive self-improvement through its rule memory. The key idea is to feed navigation knowledge distilled from past successes and failures back into subsequent decisions. The updated rules guide later navigation, whose trajectories and outcomes in turn provide feedback for further rule updates. To this end, we introduce a retrospectively updated agentic rule memory. After each episode, the agent distills trajectory and visual observations into reusable navigation rules and updates the memory across test-time episodes, which allows the accumulated knowledge to be refined using subsequent outcomes. To use this knowledge during exploration, we retrieve rules according to a UCB score. The UCB score reflects accumulated semantic credit, historical success rate, and an exploration bonus. As a result, experience from completed episodes is selectively fed back into subsequent navigation

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.

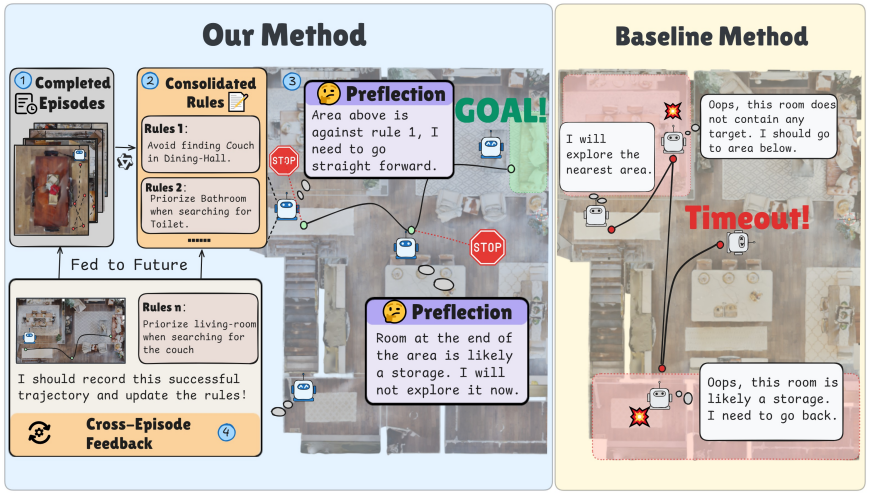

Figure 1: Comparison between RSI-Nav (left), which performs RSI across episodes and preflects before action, and a reactive baseline (right).

decisions. We further introduce memory-guided preflection. Instead of correcting mistakes after actions are executed, the agent evaluates candidate waypoints before acting. It retrieves rules distilled from past successes and failures to anticipate future observations and potential risks. This process enables the agent to filter out low-value or high-risk actions in advance, leading to more efficient and goal-directed exploration in unseen environments.

Our main contributions are summarized as follows:

- We introduce recursive self-improvement for TF-OGN through a retrospectively updated agentic rule memory. UCB-based retrieval combines semantic credit and historical success to guide future decisions.
- We propose prospective preflection guided by memory, replacing passive correction with proactive risk avoidance to reduce costly physical trial and error.
- Experiments show that our method achieves higher reported performance than the compared training-free baselines, including a 10.1% relative SR gain on MP3D.

2 Related Work

2.1 Object Goal Navigation

Early OGN approaches relied on end-to-end RL (Ye & Yang, 2021) or modular learning (Chaplot et al., 2020). RL-based methods map observations to actions but suffer from poor generalization and sim-to-real gaps. Modular methods reduce training costs with semantic maps but struggle to generalize to unseen environments without task-specific training.

Recently, LLMs (Singh & et al., 2025; Bai et al., 2023) and VLMs (Li et al., 2023) have been introduced to enable training-free OGN (Kuang et al., 2024). Methods like ZSON (Majumdar et al., 2022), CoW (Gadre et al., 2023), and VLFM (Yokoyama et al., 2024) utilize VLMs to evaluate semantic similarity between frontiers and the target to guide exploration. Meanwhile, approaches like L3MVN (Yu et al., 2023) and SG-Nav (Yin et al., 2024) leverage LLMs and 3D scene graphs for high-level spatial reasoning and long-horizon planning. To further improve efficiency and robustness in complex real-world scenarios, recent methods improve adaptive exploration (Zhang et al., 2025), multimodal scene representations (Huang et al., 2025), and cross-floor planning (Gong et al., 2026). However, many of the compared TF-OGN methods rely on priors fixed at deployment and do not explicitly maintain memory conditioned on outcomes across episodes.

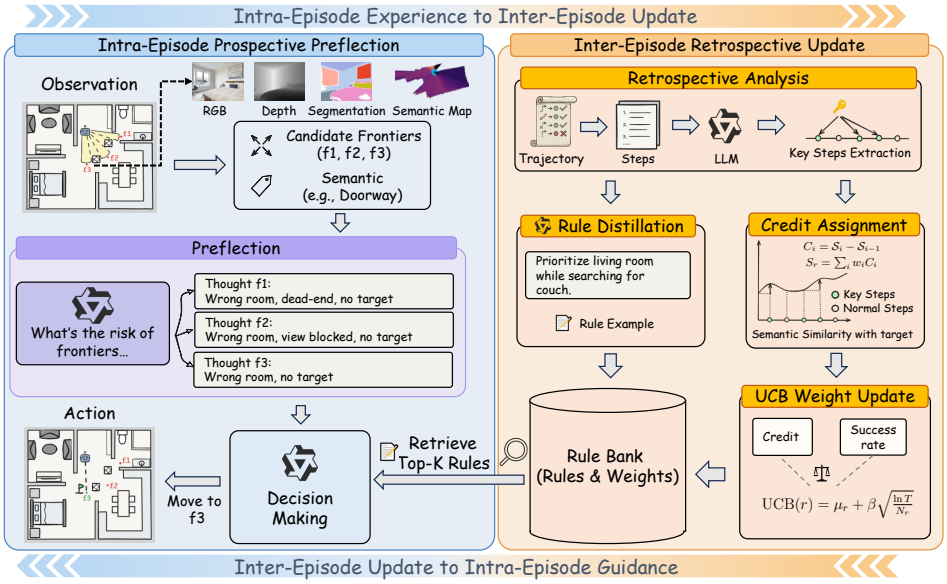

Figure 2: Method Overview of RSI-Nav, implementing RSI through rule-memory updates and reuse. ***Left (Intra-Episode Prospective Preflection):*** The agent retrieves top-K rules to evaluate frontier risks before acting. ***Right (Inter-Episode Retrospective Update):*** After each navigation episode, the framework distills new rules from the trajectory and updates rule utilities for UCB-based retrieval in future episodes.

2.2 Self-Improvement in Agents

Experience reuse in agents has gained attention. Relevant directions include prompt and memory optimization. For prompt optimization, methods (Lin et al., 2024; Wang et al., 2023; Yao et al., 2023; Xiang et al., 2025; Wu et al., 2024) aim to automatically search for high-quality instructions to guide language models toward producing more accurate outputs. For instance, PromptAgent (Wang et al., 2023) formulates prompt optimization as a strategic planning problem and leverages Monte Carlo tree search to efficiently navigate the expert-level prompt space. Similarly, OPRO (Yang et al., 2023) generates new instructions by prompting the model with previously generated candidates to iteratively refine prompt quality.

On the other hand, memory optimization methods (Zhong et al., 2024; Chhikara et al., 2025; Wang et al., 2024b; Chen et al., 2025) focus on managing contextual information and past experiences to support long-term decision-making and consistency. Reflexion (Shinn et al., 2023) enables agents to reflect on task feedback and store episodic insights, which promotes continuous self-improvement over time without weight updates. Concurrently, MemoryBank (Zhong et al., 2024) hierarchically summarizes events and dynamically updates memory based on recency and relevance to mitigate forgetting and enhance long-term retention. Despite these advancements, memory conditioned on outcomes across navigation episodes remains underexplored in OGN. Many of the compared TF-OGN frameworks do not explicitly reuse completed outcomes in later episodes, which can lead to repeated errors and limited adaptation.

3 Methodology

3.1 Problem Formulation

In this paper, we focus on the training-free object-goal navigation (TF-OGN) task. Unlike learning-based paradigms that rely on massive environment interactions to learn navigation policies, the training-free paradigm deploys off-the-shelf foundation models with all parameters frozen, requiring the agent to navigate in unstructured environments and locate an instance of a specific target object category c (e.g., "couch") without any task-specific fine-tuning.

Formally, we follow the common practice (Zhang et al., 2025; Huang et al., 2025; Gong et al., 2026) for the TF-OGN task evaluation. An episode begins with the agent placed at an initial pose p_0 within an unexplored scene. At any given time step t, the agent captures an egocentric RGB-D image $I_t = \langle I_t^{rgb}, I_t^{depth} \rangle$ and obtains its spatial tracking pose $p_t = (x_t, y_t, \theta_t)$. Relying on these sensory inputs alongside its maintained memory state, the agent determines and performs a discrete action $a_t \in \mathcal{A}$. Specifically, the permissible action space $\mathcal{A}$ is defined as a set of six distinct commands: `MOVE_FORWARD`, `TURN_LEFT`, `TURN_RIGHT`, `LOOK_UP`, `LOOK_DOWN`, and `STOP`. A navigation episode ends when the agent executes the `STOP` action or exceeds the maximum step limit $T_{\max}$. The task succeeds if the agent stops within 1 meter of the visible target object c. Otherwise, the episode is considered a failure.

3.2 Overview

Under tight step budgets, agents should reuse past outcomes to reduce physical trial and error. Without fine-tuning, our framework performs RSI by coupling online spatial exploration with a rule memory across episodes. Figure 2 shows two phases: prospective preflection within an episode and retrospective memory updates across episodes:

Intra-Episode Prospective Preflection. Within a navigation episode, the agent actively constructs a 2D environment map and extracts candidate frontiers. Before moving, the system retrieves past empirical rules from our constructed memory for the LLM. It then performs "preflection" to evaluate the potential risks of each frontier and select the optimal exploration direction.

Inter-Episode Retrospective Update. After each episode, the agent retrospectively updates the rule memory. It uses the LLM to review the trajectory, identify key decisions, and distill them into generalizable navigation rules. To manage this growing rule bank, we apply a UCB algorithm to dynamically balance the exploitation of proven rules and the exploration of new ones.

3.3 Spatial Mapping and Low-Level Control

To effectively navigate in unknown and unstructured environments, the agent must first establish a geometric understanding of the surrounding physical space and possess reliable low-level locomotion capabilities. At each time step t, the agent leverages egocentric RGB-D observations and odometry to incrementally project 3D depth data into a global coordinate frame, dynamically maintaining a 2D Occupancy Grid Map. By mapping 3D data onto a 2D plane, this projection reduces computational complexity while retaining critical information regarding traversable areas and obstacle positioning.

Based on this occupancy map, the system continuously extracts a set of candidate frontiers. A frontier is defined as the geometric boundary separating known free space from unexplored regions. By explicitly identifying these frontiers, the agent can systematically target the most informative edges of its current knowledge, ensuring that exploration is driven by spatial uncertainty rather than random wandering.

Once a target coordinate is determined by the LLM planner (see the preflection section) or identified as the target object c by a pre-trained vision model, it becomes a 2D local waypoint. A low-level navigation controller (Zhao et al., 2021) then navigates the agent to this waypoint. This hierarchical design enhances system efficiency by selectively invoking the LLM only when necessary, thereby minimizing redundant API calls. To prevent the agent from getting stuck, the controller executes movement for a maximum of H steps. Execution stops early if the frontier is fully explored, the waypoint becomes unreachable, or a visual trigger starts a new decision cycle. This step bound ensures the agent stays reactive to new observations and recovers quickly from failures.

3.4 Retrospective Agentic Rule Memory

In training-free navigation, model parameters remain frozen, so our agent retains experience across episodes in an external rule memory. To enable RSI without fine-tuning, our framework introduces a post-episode reflection mechanism that distills low-level trajectory data into reusable navigation rules. Regardless of whether an episode ends in success or failure, the system compiles the agent's complete history of frontier decisions and executes a two-stage rule generation pipeline. First, the LLM reviews the global trajectory to isolate informative decision steps associated with the episode's outcome.

Table 1: **Comparisons with SOTA methods.** The table compares learning-based (top, ✗) and training-free methods (bottom, ✓) on the HM3D and MP3D datasets across Success Rate (SR) and Success weighted by Path Length (SPL). All baseline results are taken from their original publications. Arrow "↑" denotes higher is better.

Method	Venue	Training-free	Instruction Interpolator		HM3D		MP3D	
			Vision	Language	SR ↑	SPL ↑	SR ↑	SPL ↑
RIM (Chen et al., 2023)	IROS'23	✗	-	-	57.8	27.2	50.3	17.0
OVG-Nav (Yoo et al., 2024)	RAL'24	✗	-	-	-	-	35.8	12.3
VLFM (Yokoyama et al., 2024)	ICRA'24	✗	BLIP-2	-	52.5	30.4	36.4	17.5
PIRLNav (Ramrakhya et al., 2023)	CVPR'23	✗	-	-	64.1	27.1	-	-
XGX (Wasserman et al., 2024)	ICRA'24	✗	-	-	72.9	35.7	-	-
ZSON (Majumdar et al., 2022)	NeurIPS'22	✓	CLIP	-	25.5	12.6	15.3	4.8
L3MVN (Yu et al., 2023)	IROS'23	✓	-	RoBERTa-large	50.4	23.1	34.9	14.5
PixNav (Cai et al., 2024)	ICRA'24	✓	LLaMA-Adapter	GPT-4	37.9	20.5	-	-
VLFM (Yokoyama et al., 2024)	ICRA'24	✓	BLIP-2	-	50.9	23.6	32.5	15.9
SG-Nav (Yin et al., 2024)	NeurIPS'24	✓	LLaVA-1.6-7B	GPT-4	54.0	24.9	40.2	16.0
ActPept (Guo et al., 2024)	RAL'24	✓	GraphSAGE	-	-	-	39.8	17.4
InstructNav (Long et al., 2024)	CoRL'24	✓	GPT-4V	GPT-4	58.0	20.9	-	-
ApexNav (Zhang et al., 2025)	RAL'25	✓	BLIP-2	DeepSeek-V3	59.6	33.0	39.2	17.8
MFNP (Zhang et al., 2024)	ICRA'25	✓	Qwen-VLChat-Int4	Qwen2-7B	58.3	26.7	41.1	15.4
MSGNav (Huang et al., 2025)	CVPR'26	✓	GPT-4o	GPT-4o	63.0	31.4	-	-
ASCENT (Gong et al., 2026)	RAL'26	✓	BLIP-2	Qwen2.5-7B	65.4	33.5	44.5	15.5
RSI-Nav (Ours)	-	✓	BLIP-2	Qwen3-8B	**67.3**	**33.9**	**49.0**	**19.1**

Subsequently, a secondary LLM query analyzes the local observations and decision contexts of these specific steps to synthesize reusable navigation rules. Finally, these newly formulated rules are deposited into the rule memory bank $\mathcal{M}$ for future invocation and dynamic weight evaluation.

3.4.1 Semantic-driven Credit Assignment

Distilling high-quality rules from a lengthy exploration trajectory requires estimating which actions contributed to the final outcome. We must therefore assign a credit to each step, using semantic progress to estimate its contribution. Because a single episode can encompass dozens of steps, simply treating all actions equally is ineffective. Inspired by information potential theory (Xie et al., 2026), we propose a credit assignment mechanism based on sequential improvements in semantic similarity. For the i-th step in the trajectory, we calculate a composite semantic similarity score $\mathcal{S}_i$ between the currently observed scene and the target object, reflecting the potential value of that step:

$$\mathcal{S}_i = \alpha \cdot \mathcal{S}_{\text{room}}^{(i)} + (1-\alpha) \cdot \mathcal{S}_{\text{obj}}^{(i)}, \tag{1}$$

where $\mathcal{S}_{\text{room}}^{(i)}$ and $\mathcal{S}_{\text{obj}}^{(i)}$ represent the cosine similarities of the currently observed room type and the visible object set against the target category, respectively, and α serves as a balancing coefficient. The initial credit C_i assigned to the i-th step is then defined as the marginal gain in semantic similarity relative to the preceding step:

$$C_i = \mathcal{S}_i - \mathcal{S}_{i-1}. \tag{2}$$

To differentiate the reward distribution between successful and failed episodes, we apply a clipping operation to C_i to prevent extreme values and compute a normalized weight w_i using a softmax function. Ultimately, the initial support score S_r assigned to the newly generated rule r is defined as the weighted sum of the contributions from its associated key steps:

$$S_r = \sum_i w_i \cdot C_i. \tag{3}$$

3.4.2 Dynamic Memory Management via UCB

As the agent continuously explores multiple environments, the rule memory bank naturally expands. Given the high cost of physical trial and error, the system must balance the exploitation of high-utility rules with the exploration of newly formulated, unverified ones. To mathematically formalize this trade-off, we treat rule retrieval as a multi-armed bandit problem and employ a UCB management strategy.

For any rule $r \in \mathcal{M}$, its utility estimate μ_r summarizes both semantic credit and historical success. This utility is iteratively smoothed using a momentum-based update that incorporates the newly

computed semantic credit S_r and historical success rate p_r:

$$\mu_r \leftarrow \eta \cdot \mu_r + (1 - \eta) \cdot (S_r + p_r), \tag{4}$$

where $\eta \in [0, 1]$ serves as a momentum decay coefficient, and p_r is the historical success rate of rule r. Prior to initiating a new episode, the framework computes the UCB score for all rules residing in the memory bank:

$$\text{UCB}(r) = \mu_r + \beta\sqrt{\frac{\ln T}{N_r}}, \tag{5}$$

where T represents the total number of exploration episodes the agent has completed thus far, N_r is the historical frequency with which rule r has been retrieved and applied, and β is a hyperparameter modulating the exploration degree. In this design, semantic credit S_r and historical success rate p_r jointly update the exploitation estimate μ_r, while the confidence bonus in Eq. 5 encourages the validation of under-explored rules; semantic credit assignment, success feedback, and UCB are thus sequential components of a single rule-scoring mechanism.

Crucially, to prioritize newly generated rules for initial use, their UCB scores are initialized to infinity (∞) when $N_r = 0$. Finally, the memory bank is sorted in descending order based on these computed $\text{UCB}(r)$ scores. The top-K rules are extracted and injected as experiential priors into the LLM's context during the preflection phase (as detailed below), feeding consolidated experience into the prospective phase and supporting the next RSI cycle.

3.5 Prospective Memory-Guided Preflection

As described in Section 3.3, the mapping module identifies a set of candidate frontiers at each decision step. In environments with constrained step budgets, relying on greedy distance-based strategies or static commonsense priors often traps the agent in local optima or leads to repetitive exploration of irrelevant regions. To circumvent these prohibitively expensive physical trial-and-error costs, we propose an LLM-based memory-guided preflection mechanism. This module enables the agent to proactively anticipate and avoid potential failures before executing actions that consume the limited step budget.

To implement this, we construct a structured context prompt that integrates geometric distances with local semantic observations, such as visible objects and spatial layouts for each candidate. Alongside these current observations, our approach dynamically injects the agent's self-summarized decision memory into this reasoning context.

Specifically, at each decision stage, the system queries a retrospectively updated rule bank $\mathcal{M}$ (Section 3.4) to retrieve the top-K experiential rules ranked by UCB. To balance the exploitation of reliable past rules with the exploration of newly formed insights, the retrieval is governed by an Upper Confidence Bound (UCB) selection criterion:

$$\mathcal{R}_{\text{top}K} = \text{TopK}_{r \in \mathcal{M}} \big(\text{UCB}(r)\big), \tag{6}$$

where $\text{UCB}(r)$ denotes the confidence score of rule r. During preflection, the LLM assesses how the retrieved rules apply to the current frontier observations.

To encourage structured risk assessment and reduce impulsive decisions, we replace direct target selection with an explicit reasoning process. The LLM must first perform failure prediction for each candidate frontier by synthesizing injected rules with current observations. For example, it might predict a high risk of failure if a frontier's visual cues conflict with established spatial rules. This structured assessment allows the LLM to compare predicted failure risks with exploration rewards when selecting a frontier.

4 Experiment

4.1 Experimental Setup

4.1.1 Datasets and Evaluation Metrics

We evaluate our proposed method on two standard ObjectNav benchmarks derived from the Habitat Challenge. Specifically, we utilize the HM3D-Semantics-v0.1 dataset from the 2022 Habitat challenge (Yadav et al., 2023) and the updated MP3D dataset from the 2021 Habitat challenge (Chang

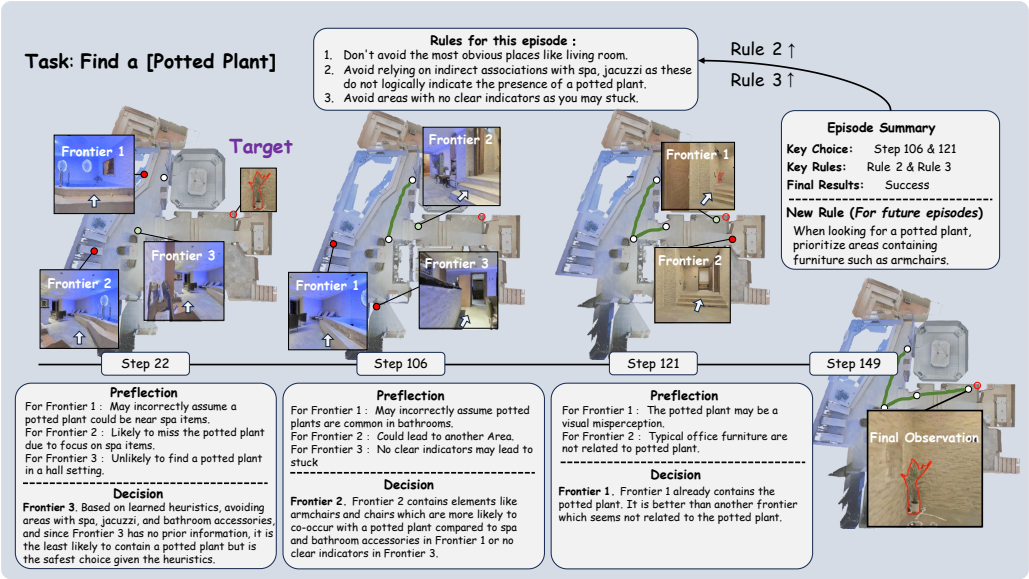

Figure 3: Navigation process visualization of RSI-Nav. We provide the agent's trajectory alongside its reasoning process. During the episode, the agent utilizes prospective preflection to evaluate frontiers and proactively avoid unpromising rooms. After the episode, RSI-Nav generates a new rule from the history and updates its Rule Bank. This feedback supports RSI by connecting short-term navigation outcomes to future decisions.

et al., 2017). Following the standard evaluation protocol established in prior ObjectNav methods (Gong et al., 2026), we employ Success Rate (SR) and Success weighted by Path Length (SPL) to comprehensively evaluate the performance of our method.

4.1.2 Baselines

To evaluate the effectiveness of our proposed approach, we compare it against a comprehensive set of state-of-the-art baselines, which are broadly categorized into learning-based and training-free methods. The learning-based category includes RIM (Chen et al., 2023), OVG-Nav (Yoo et al., 2024), PIRLNav (Ramrakhya et al., 2023), and XGX (Wasserman et al., 2024), which heavily rely on task-specific training architectures to achieve navigation. In the training-free category, we evaluate foundational methods such as ZSON (Majumdar et al., 2022) and ActPept (Guo et al., 2024), which utilize contrastive models like CLIP or graph-based structures for semantic guidance, alongside progressive frameworks that incorporate LLMs and VLMs for explicit instruction interpolation. Specifically, we compare against L3MVN (Yu et al., 2023), PixNav (Cai et al., 2024), SG-Nav (Yin et al., 2024), and InstructNav (Long et al., 2024), which use different language or vision-language models to interpret language and guide exploration. Furthermore, we benchmark against recent advanced training-free frameworks including VLFM (Yokoyama et al., 2024) and ApexNav (Zhang et al., 2025), which employ BLIP-2 (Li et al., 2023), with ApexNav additionally using DeepSeek-V3 (Liu et al., 2024), as well as the latest state-of-the-art methods such as MFNP (Zhang et al., 2024), ASCENT (Gong et al., 2026), and MSGNav (Huang et al., 2025), with MFNP and ASCENT using the Qwen model family (Bai et al., 2023; Yang et al., 2024a;b; 2025), and MSGNav using GPT-4o.

4.1.3 Implementation Details

For the multimodal reasoning backbone, we employ the pre-trained BLIP-2 (Li et al., 2023) as our vision foundation model and Qwen3-8B (Yang et al., 2025) as the language model instruction interpolator. All model parameters remain frozen; prospective preflection and retrospective rule updates operate only through inference-time prompting and UCB updates. Following established evaluation protocols from prior works (Batra et al., 2020; Huang et al., 2025; Gong et al., 2026), we set the success distance thresholds to 1.0 meters for both benchmarks. The LLM uses deterministic decoding

Table 2: Ablation of preflection and rule updates.

No.	Module		HM3D		MP3D	
	Prospective	Retrospective	SR ↑	SPL ↑	SR ↑	SPL ↑
1			64.7	32.5	43.9	15.8
2	✓		66.5	33.5	47.4	18.4
3		✓	66.7	33.6	48.3	18.7
4	✓	✓	**67.3**	**33.9**	**49.0**	**19.1**

Table 3: Ablation on the LLM backbone.* denotes results obtained by reproducing the experiments.

Method	LLM backbone	HM3D		MP3D	
		SR ↑	SPL ↑	SR ↑	SPL ↑
ASCENT*	Qwen2.5-7B	65.2	32.8	44.9	16.1
	Qwen3-8B	65.3	33.0	45.3	16.2
RSI-Nav	Qwen2.5-7B	67.0	33.9	48.8	19.0
	Qwen3-8B	**67.3**	**33.9**	**49.0**	**19.1**

(temperature 0); the cross-episode memory protocol is specified below. Additional hyperparameter configurations, computational infrastructure, and further implementation details are provided in Appendix A.

4.1.4 Cross-Episode Evaluation Protocol

Episodes are processed sequentially in the official benchmark order. During episode t, the agent can access only rules distilled from completed episodes $1, \ldots, t-1$, and never from the current or future episodes. All VLM and LLM parameters remain frozen and only the external, non-parametric rule memory is updated, which is initialized empty and reset independently for each benchmark dataset. The post-episode update uses the completed trajectory, visual observations, task-provided goal category, and binary episode outcome from Habitat; it does not access ground-truth target locations, instance annotations, shortest paths, or future observations.

4.2 Quantitative Evaluation

Table 1 shows that RSI-Nav achieves the best training-free SR and SPL on both HM3D and MP3D. Our method achieves the best reported results among the compared training-free methods on both datasets, consistently outperforming existing training-free approaches. On the HM3D dataset, our method reaches an SR of 67.3% and an SPL of 33.9%, surpassing ASCENT by 1.9 percentage points in SR and 0.4 percentage points in SPL. On MP3D, RSI-Nav achieves 49.0% SR and 19.1% SPL. Relative to the strongest prior result for each metric (44.5% SR from ASCENT and 17.8% SPL from ApexNav), these values improve SR by 4.5 percentage points (a 10.1% relative gain) and SPL by 1.3 percentage points. For a controlled backbone comparison, Table 3 evaluates RSI-Nav and ASCENT with the same Qwen2.5-7B model; our method still leads.

The stronger MP3D result suggests that reusing experience across episodes helps in complex unseen layouts. The SPL improvements further indicate that preflection reduces redundant exploration.

4.3 Ablation Studies

4.3.1 Effectiveness of Prospective Preflection

To verify the effectiveness of the proposed preflection, we disable it during inference. As shown in Table 2, integrating the preflection module without rule memory (Setting 2) yields a noticeable performance boost over the baseline (Setting 1), improving the SR by 1.8 percentage points on HM3D and 3.5 percentage points on the more challenging MP3D dataset. Notably, we also observe a steady increase in SPL. These results are consistent with the hypothesis that prior-to-action anticipation helps reduce physical trial and error under a limited step budget. By requiring the LLM to predict frontier risks before taking actions that consume the step budget, the agent filters out deceptive local optima and avoids dead ends. This proactive risk avoidance conserves the step budget and yields more purposeful trajectories. With rule memory enabled, adding preflection (Setting 4 versus 3) increases SR by 0.6 and 0.7 percentage points on HM3D and MP3D, respectively.

4.3.2 Effectiveness of Retrospective Feedback

To evaluate retrospective feedback, we isolate the rule memory and its UCB update (Setting 3 in Table 2). Compared with the static baseline, this setting improves SR by 2.0 percentage points on HM3D and 4.4 percentage points on MP3D. The larger MP3D gain is consistent with fixed priors

being less reliable in complex or unusual layouts. Together, semantic credit assignment and rule updates feed completed outcomes into future decisions without changing model parameters. With preflection enabled, adding rule memory (Setting 4 versus 2) increases SR by 0.8 and 1.6 percentage points on HM3D and MP3D, respectively.

4.3.3 Effectiveness of LLM Backbone

To investigate the influence of the language foundation model on the overall navigation performance, we conduct an ablation study by changing the LLM backbone within the Instruction Interpolator, while keeping the vision foundation model fixed to BLIP-2. As presented in Table 3, we evaluate the framework using Qwen2.5-7B and Qwen3-8B. Within the tested Qwen2.5-7B and Qwen3-8B configurations, RSI-Nav shows only small performance differences on both datasets. This supports robustness to these two tested backbones but does not establish a general relationship between language model scale and navigation performance. Meanwhile, RSI-Nav explicitly grounds the agent's decision-making in a dynamic rule memory. During the preflection phase, the LLM is injected with structured local observations alongside high-value historical rules retrieved via the UCB selection criterion. A detailed comparison of LLM calls, latency, and token usage is provided in Appendix B.

4.3.4 Ablation on Rule Retrieval Strategy

Complementary to the module ablation above, we isolate the contribution of the retrieval strategy on MP3D (Table 4). All rows here keep preflection enabled and vary only how rules are selected. Semantic only ranks rules by S_r, while UCB only uses historical success with the exploration bonus. FULL combines both signals in the utility update before UCB ranking. Naive retrieval (random rules or FIFO) brings little gain over retrieving no rule, while trajectory-derived semantic credit and UCB-based utility each help on their own. Combining them yields the best result.

Table 4: Ablation on rule retrieval strategies (MP3D). Combining UCB with trajectory-derived semantic credit outperforms either signal alone and naive baselines.

Retrieval strategy	**SR** ↑	**SPL** ↑
Preflection only (no rules)	47.4	18.4
Random rules	47.5	18.2
FIFO (recency only)	48.0	18.5
Semantic only	48.2	18.6
UCB only	48.7	18.9
FULL (UCB + semantic)	**49.0**	**19.1**

4.4 Qualitative Evaluation

To intuitively demonstrate the decision-making ability of RSI-Nav, we provide a qualitative case study from the MP3D dataset, as shown in Figure 3. The agent is tasked with finding a <Potted Plant> in a multi-room layout. Instead of greedily exploring the nearest frontiers, RSI-Nav uses retrieved rules to evaluate frontiers. For instance, at Step 22, the agent triggers the preflection module before physical movement. By retrieving historical rules from the Rule Bank (e.g., Rule 2: "Avoid relying on indirect associations with spa, jacuzzi"), the LLM anticipates the failure risk of Frontier 1 and successfully bypasses it. Later at Step 106, preflection correctly guides the agent to prioritize an area with armchairs, anticipating a higher co-occurrence probability with the target. After the episode ends, RSI-Nav feeds its physical-interaction outcomes back into the rule memory. Following the trajectory, RSI-Nav updates the utility estimates of the applied rules (Rules 2 and 3) using episode feedback. Furthermore, the LLM distills a new heuristic from this experience: "When looking for a potted plant, prioritize areas containing furniture such as armchairs" enriching the Rule Bank to guide future cross-scene explorations. More qualitative comparison is provided in Appendix G.

5 Conclusion

We propose RSI-Nav, a training-free framework that performs RSI through rule-memory updates and reuse in TF-OGN. Our approach introduces a retrospectively updated rule memory that extracts actionable knowledge from past trajectories and integrates trajectory-derived semantic credit and historical success into UCB-based retrieval. To minimize physical trial-and-error, we develop a memory-guided preflection module that shifts the exploration paradigm from passive correction to proactive risk avoidance. Across HM3D and MP3D, RSI-Nav achieves higher reported performance than the compared training-free baselines.

AI USE STATEMENT

We used generative AI tools to assist with writing and grammar checking. Within the navigation agent, AI models were used to generate high-level action selections for the task. The authors reviewed the manuscript structure and implemented code-level constraints to restrict the AI-generated action choices to reasonable options. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

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. *arXiv preprint arXiv: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 (ICRA)*, pp. 5228–5234. IEEE, 2024.

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

Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020.

Nuo Chen, Hongguang Li, Jianhui Chang, Juhua Huang, Baoyuan Wang, and Jia Li. Compress to impress: Unleashing the potential of compressive memory in real-world long-term conversations. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 755–773, 2025.

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.

Shizhe Chen, Thomas Chabal, Ivan Laptev, and Cordelia Schmid. Object goal navigation with recursive implicit maps. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7089–7096. IEEE, 2023.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Ronghao Dang, Liuyi Wang, Zongtao He, Shuai Su, Jiagui Tang, Chengju Liu, and Qijun Chen. Search for or navigate to? dual adaptive thinking for object navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8250–8259, 2023.

Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23171–23181, 2023.

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*, 2026.

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.

Xun Huang, Shijia Zhao, Yunxiang Wang, Xin Lu, Wanfa Zhang, Rongsheng Qu, Weixin Li, Yunhong Wang, and Chenglu Wen. Msgnav: Unleashing the power of multi-modal 3d scene graph for zero-shot embodied navigation. *arXiv preprint arXiv:2511.10376*, 2025.

Jinyeon Kim, Cheolhong Min, Byeonghwi Kim, and Jonghyun Choi. Pre-emptive action revision by environmental feedback for embodied instruction following agents. In *CoRL*, 2024.

Yuxuan Kuang, Hai Lin, and Meng Jiang. Openfmnav: Towards open-set zero-shot object navigation via vision-language foundation models. *arXiv preprint arXiv:2402.10670*, 2024.

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 *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Xiaoqiang Lin, Zhongxiang Dai, Arun Verma, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. Prompt optimization with human feedback. *arXiv preprint arXiv:2405.17346*, 2024.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. In *Conference on Robot Learning*, 2024.

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

Ram Ramrakhya, Dhruv Batra, Erik Wijmans, and Abhishek Das. Pirlnav: Pretraining with imitation and rl finetuning for objectnav. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17896–17906, 2023.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9339–9347, 2019.

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.

Aaditya Singh and Adam Fry et al. Openai gpt-5 system card, 2025. URL `https://arxiv.org/abs/2601.03267`.

Fuchun Sun, Runfa Chen, Tianying Ji, Yu Luo, Huaidong Zhou, and Huaping Liu. A comprehensive survey on embodied intelligence: Advancements, challenges, and future perspectives. *CAAI Artificial Intelligence Research*, 3:9150042, 2024.

Weihao Tan, Wentao Zhang, Shanqi Liu, Longtao Zheng, Xinrun Wang, and Bo An. True knowledge comes from practice: Aligning large language models with embodied environments via reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=hILVmJ4Uvu`.

Hanlin Wang, Chak Tou Leong, Jian Wang, and Wenjie Li. E^2CL: Exploration-based error correction learning for embodied agents. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7626–7639, Miami, Florida, USA, November 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.448.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric P Xing, and Zhiting Hu. Promptagent: Strategic planning with language models enables expert-level prompt optimization. *arXiv preprint arXiv:2310.16427*, 2023.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. *arXiv preprint arXiv:2409.07429*, 2024b.

Justin Wasserman, Girish Chowdhary, Abhinav Gupta, and Unnat Jain. Exploitation-guided exploration for semantic embodied navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2901–2908. IEEE, 2024.

Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. *arXiv preprint arXiv:1911.00357*, 2019.

Yurong Wu, Yan Gao, Bin Benjamin Zhu, Zineng Zhou, Xiaodi Sun, Sheng Yang, Jian-Guang Lou, Zhiming Ding, and Linjun Yang. Strago: Harnessing strategic guidance for prompt optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10043–10061, 2024.

Jinyu Xiang, Jiayi Zhang, Zhaoyang Yu, Fengwei Teng, Jinhao Tu, Xinbing Liang, Sirui Hong, Chenglin Wu, and Yuyu Luo. Self-supervised prompt optimization. *arXiv preprint arXiv:2502.06855*, 2025.

Yutao Xie, Nathaniel Thomas, Nicklas Hansen, Yang Fu, Li Erran Li, and Xiaolong Wang. Tips: Turn-level information-potential reward shaping for search-augmented llms. In *The Fourteenth International Conference on Learning Representations*, 2026.

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

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *eprint arXiv: 2407.10671*, 2024a.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*, 2023.

Zebin Yang, Sunjian Zheng, Tong Xie, Tianshi Xu, Bo Yu, Fan Wang, Jie Tang, Shaoshan Liu, and Meng Li. Efficientnav: Towards on-device object-goal navigation with navigation map caching and retrieval. *Advances in Neural Information Processing Systems*, 38:4286–4312, 2026.

Weiran Yao, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh Murthy, Zeyuan Chen, Jianguo Zhang, Devansh Arpit, et al. Retroformer: Retrospective large language agents with policy gradient optimization. *arXiv preprint arXiv:2308.02151*, 2023.

Xin Ye and Yezhou Yang. Efficient robotic object search via hiem: Hierarchical policy learning with intrinsic-extrinsic modeling. *IEEE robotics and automation letters*, 6(3):4425–4432, 2021.

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. *Advances in Neural Information Processing Systems*, 37:5285–5307, 2024.

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)*, pp. 42–48. IEEE, 2024.

Hwiyeon Yoo, Yunho Choi, Jeongho Park, and Songhwai Oh. Commonsense-aware object value graph for object goal navigation. *IEEE Robotics and Automation Letters*, 9(5):4423–4430, 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 (IROS)*, pp. 3554–3560. IEEE, 2023.

Lingfeng Zhang, Hao Wang, Erjia Xiao, Xinyao Zhang, Qiang Zhang, Zixuan Jiang, and Renjing Xu. Multi-floor zero-shot object navigation policy. *arXiv preprint arXiv:2409.10906*, 2024.

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*, 2025.

Xiaoming Zhao, Harsh Agrawal, Dhruv Batra, and Alexander Schwing. The Surprising Effectiveness of Visual Odometry Techniques for Embodied PointGoal Navigation. In *Proc. ICCV*, 2021.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 19724–19731, 2024.

This appendix provides the implementation details, hyperparameters, inference-cost analysis, retrieval-strategy analysis, qualitative rule examples, failure analysis, and prompt templates referenced by the main paper. Consistently with the main paper, we frame the task as training-free Object-Goal Navigation in which RSI-Nav performs RSI through its rule memory: all foundation-model parameters remain frozen, while an external, non-parametric rule memory is updated *retrospectively* from completed episodes and then reused to guide *prospective* pre-action decisions.

A Implementation Details

We evaluate RSI-Nav in the Habitat ObjectNav simulator under the standard training-free setting; all model weights remain frozen at test time. The stack uses Habitat-Sim v0.3.1 and Habitat-Lab v0.3.1 on Python 3.9, on the HM3D-Semantics-v0.1 dataset (2,000 episodes, 20 scenes, 6 goal categories) and the MP3D ObjectNav validation split (2,195 episodes, 11 scenes, 21 goal categories).

Episodes follow the official benchmark order. During episode t, the rule bank contains only rules distilled from completed episodes $1, \ldots, t-1$; it is initialized empty and reset independently for each dataset. The post-episode update uses the completed trajectory, visual observations, task-provided goal category, and binary outcome, but not ground-truth target locations, instance annotations, shortest paths, or future observations.

A.1 Perception and mapping

The agent receives egocentric RGB-D observations at 640×480 with a 79° horizontal field of view; depth is clipped to $[0.5, 5.0]$ meters. The camera is mounted at 0.88 m (pose offset $[0, 0.88, 0]$), and the agent body has height 0.88 m and radius 0.18 m; simulator sliding is disabled for determinism. The discrete action space is `MOVE_FORWARD` (0.25 m), `TURN_LEFT/RIGHT` and `LOOK_UP/DOWN` (30° each), and `STOP`, with a 500-step episode limit; success requires stopping within 1.0 m of a visible target. Open-vocabulary perception combines category-level detection with mask refinement (confidence thresholds 0.8 for COCO categories and 0.4 otherwise), and scene type is classified from RGB to supply room cues. The agent maintains a multi-channel global map of 1600×1600 pixels at 20 pixels/meter (80 m $\times$ 80 m), with a vertical obstacle filter in $[0.61, 0.88]$ m, an obstacle-area threshold of $1.5\,\mathrm{m}^2$, a hole-filling threshold of 10^5 pixels, a morphological erosion size of 5, and a 0.1 m semantic voxel cache. Frontiers are extracted via a fog-of-war boundary (area threshold $1.5\,\mathrm{m}^2$, raycast distance 4.5 m); candidates are ranked and truncated to top-3, with a 3.0 m near-frontier shortcut and SSIM-based suppression (threshold 0.75) of duplicate views.

A.2 Foundation models

Semantic image–text alignment uses the frozen BLIP-2 ITM model. The primary configuration uses frozen Qwen3-8B for high-level reasoning, preflection, and rule generation; the backbone ablation in the main paper additionally evaluates Qwen2.5-7B. Decoding uses temperature 0, $top_p = 1.0$, $top_k = 50$, and at most 512 new tokens. Perception and reasoning models run as independent local inference servers, and no model is fine-tuned on HM3D, MP3D, or an environment-specific split. All simulation, environment rendering, and model inference run on one workstation with an AMD EPYC 7542 32-core CPU and two NVIDIA RTX A6000 GPUs.

A.3 Retrospective rule-memory hyperparameters

The composite similarity uses $\alpha = 0.5$, weighting room and object similarity equally. Step credit C_i is clipped to $[-1, 1]$ and converted to w_i by a softmax with temperature $\tau = 0.30$; critical support steps are selected with a top-3 policy. Consistently with the main paper, the combined utility is updated as $\mu_r \leftarrow \eta\mu_r + (1-\eta)(S_r + p_r)$ with momentum $\eta = 0.80$, where S_r is trajectory-derived semantic credit and p_r is the historical success rate of rule r. UCB retrieval uses $\beta = 1.0$; T is the number of completed episodes, N_r is the retrieval-and-application count of rule r, and an unseen rule receives a cold-start score of ∞ when $N_r = 0$. The top-5 rules are injected into the preflection prompt at each planning step.

B LLM INFERENCE COST

Table 5 reports the available per-episode LLM measurements. During navigation, high-level planning is triggered at frontier selections rather than at every low-level simulator step; RSI-Nav also performs the post-episode rule-generation calls described in the main paper. The available measurements do not provide a phase-wise breakdown. Relative to ASCENT, RSI-Nav uses about 10% more LLM calls in this accounting. All reported configurations use deterministic decoding (temperature 0), removing stochastic sampling as a source of variation. MSGNav* latency is estimated from token usage under the same LLM throughput.

Table 5: Available per-episode LLM cost measurements. *MSGNav latency is estimated from token usage under matched throughput.

Metric (per episode)	MSGNav*	ASCENT	RSI-Nav
LLM calls	24.8	8.7	9.6
LLM latency (s)	151.4	43.4	45.9
Avg. tokens	39.7k	9.3k	10.5k

C RETRIEVAL STRATEGY ANALYSIS

Complementing the retrieval-strategy ablation in the main paper, we analyze the number of rules K injected into the preflection prompt (Figure 4). Here "semantic" denotes the trajectory-derived credit S_r: Semantic only ranks rules by S_r, UCB only uses historical success p_r with the exploration bonus, and FULL combines S_r, p_r, and exploration through the utility and UCB updates in the main paper. In the K sensitivity study, SR is 48.8%, 49.0%, 48.4%, and 47.6% for $K = 3, 5, 10, 20$, respectively. Thus, $K = 5$ gives the best reported result in this comparison; the decline at larger K is consistent with less focused rule context.

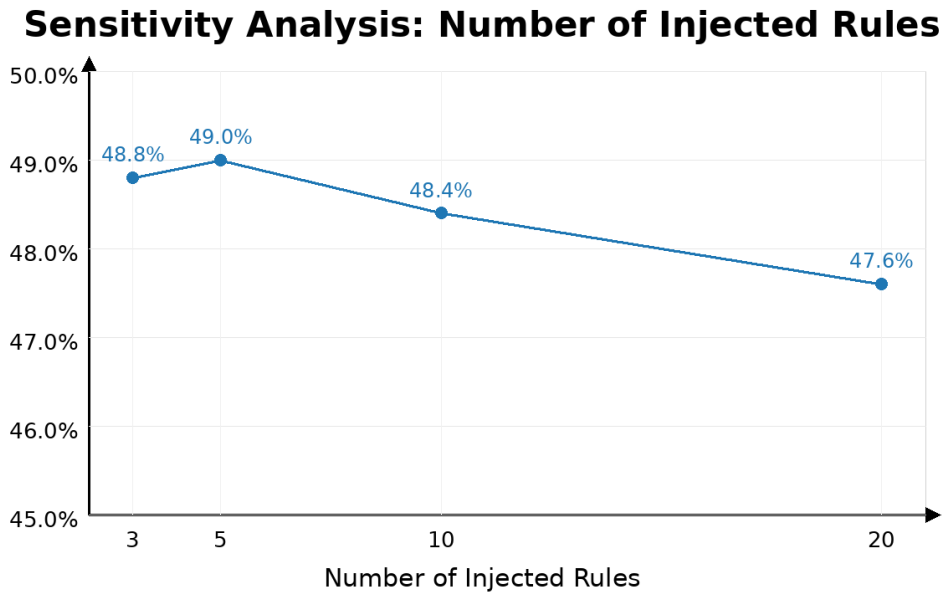

Figure 4: Success rate versus the number of retrieved rules K during preflection; performance peaks at $K = 5$ and declines as more rules are injected.

D Qualitative Examples of Evolved Rules

Table 6 presents representative rules automatically distilled from trajectory records rather than manually authored priors. The labels summarize their observed usage and retrieval histories: frequently used examples were often retrieved and applied, whereas lower-ranked examples were less often selected by top-K UCB retrieval. The latter often contain rigid constraints, such as rejecting co-occurring seating furniture or avoiding transit spaces categorically. The table entries are lightly shortened for presentation and should therefore be read as representative examples rather than verbatim model outputs or a quantitative validation of rule correctness.

Table 6: Representative generated rules, lightly shortened for presentation. "Frequently used" denotes rules often retrieved and applied; "lower-ranked" denotes examples that were less often selected by top-K UCB retrieval.

Category	Frequently used rule	Lower-ranked rule
Bed	When looking for a bed, prioritize rooms explicitly identified as bedrooms or containing sleeping furniture (pillows) because these semantic cues align with the object's function, whereas dining rooms with tables and chairs are inconsistent and should be avoided.	When looking for a bed, prioritize rooms with bedroom semantics and avoid rooms with conflicting functions (e.g., dining room, living room) even if spatially adjacent, as beds are semantically exclusive to sleeping areas.
Chair	When looking for a chair, prioritize areas containing seating furniture (like living rooms) or explicitly listed seating items (hassocks, stools) over areas with low semantic probability (like bathrooms), as chairs are functionally associated with seating zones.	When looking for a chair, avoid relying on co-occurrence with other seating furniture (like armchairs, couches, stools, or hassocks) or general room functions as primary indicators, because these cues do not guarantee the specific presence of a chair.
Couch	When looking for a couch, prioritize areas containing any seating furniture (like armchairs or chairs) because these objects share the same functional category of seating even if a dedicated couch is absent.	When looking for a couch, avoid areas containing stairs and corridors because these are semantically inconsistent with the typical living-room location of a couch.
Potted Plant	When looking for a potted plant, avoid prioritizing structural elements like beams, pillars, basements, or stone flooring; instead prioritize semantic room functions (e.g., living room, garden) or explicit furniture indicators (e.g., shelves, tables).	When looking for a potted plant, prioritize the living room because it is functionally associated with decorative greenery, whereas laundry rooms are for storage and transit spaces are for movement.
Toilet	When looking for a toilet, prioritize transit zones that lead to private rooms (e.g., bedrooms) because toilets are semantically located within private areas, not in public transit spaces.	When looking for a toilet, avoid prioritizing halls or transit furniture (balustrades, stairwells) as indicators of proximity, because these are common architectural features in living areas and do not semantically guarantee the presence of a bathroom.
TV	When looking for a TV, avoid exploring bathrooms because they are functionally unrelated to entertainment systems and have a low prior probability of containing the target.	When looking for a TV, avoid bedrooms containing only dressers and bureaus because these items indicate a sleeping area where TVs are not typically placed, unlike living rooms or entertainment centers.

E Failure Analysis

Figure 5 reports the failure reasons among failed episodes. On HM3D, **False Positive** accounts for 51.4%, followed by **Infeasible** (36.3%) and **Never Saw Target** (12.3%). On MP3D, **Infeasible** accounts for 43.7%, followed by **False Positive** (35.0%), **Never Saw Target** (10.9%), and **False Negative** (10.4%). False Positive and False Negative denote incorrect and missed target detections, respectively. Infeasible groups insufficient scene connectivity and failures to recognize or traverse stairs, while Never Saw Target denotes exhaustion of the step budget before observing the target.

These categories identify perception, reachability, and exploration bottlenecks, but this descriptive attribution does not isolate the navigation policy's contribution.

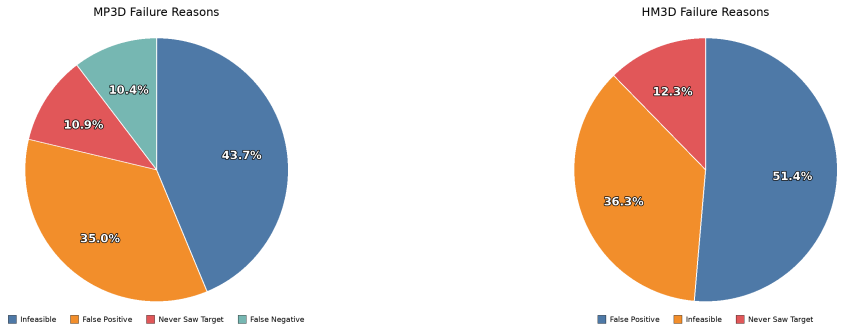

Figure 5: Failure-reason distributions among failed episodes on MP3D (*left*) and HM3D (*right*).

F Prompt Templates

Tables 7 and 8 list the prompts used for post-episode rule generation and for memory-guided frontier selection (preflection), respectively.

Table 7: Prompt for post-episode rule generation.

Post-episode Rule Generation Prompt:

```
You are a Senior Robot Logic Analyst. Your goal is to extract generalizable navigation
rules that align with Human Common Sense.
CRITICAL RULE FOR ANALYSIS: - Reject Coincidence: if success is accidental, mark as Lucky
Success (no rule). - Focus on Semantics: relate objects to room functions or furniture,
not visual noise. - Identify Anomalies: if the goal-object relation is illogical,
produce a STRICT negative rule. Respond in JSON only.
Input fields: Goal, Episode Outcome (SUCCESS/FAILURE), Auto Support Candidates, Support
Step Mode Hint, Navigation Trace (Decisions made).
Task: (1) find critical success/failure steps, (2) summarize a general rule, (3) if Lucky
output type=lucky and empty rule_content.
Return JSON: {"target":"{target}", "critical_success_steps":[...], "critical_failure_
steps":[...], "support_steps":[...], "rule_content":"When looking for {target_object},
prioritize/avoid ... because ...", "type":"positive|negative|lucky", "confidence":0.0}
```

Table 8: Prompt for memory-guided candidate-frontier selection (preflection).

Candidate Frontiers Selection (Preflection) Prompt:

```
You need to select the optimal area based on prior probabilistic data and environmental
context. Focus on navigation logic and semantics, not visual snapshot captioning. For
each candidate frontier, first predict what new observations it will reveal and identify
its potential failure mode before making the final selection. Classify reasoning into
one of: Object_Co-occurrence, Room_Function, Exploration, Backtracking, Other/Unknown.
Optional: if heuristics were injected, return Followed_Rule_Indices as 1-based indices
from the injected list.
You MUST output Preflections with exactly {num_candidates} entries (one per area). Each
reflection entry must include: Index, Expected_New_Observations, Potential_Failure_Mode.
You MUST output a valid top-level Index from 1 to {num_candidates}. Never output None,
Null, Unknown, or non-numeric Index. If uncertain, still choose the single best Index.
Respond with one JSON object only.
Input fields: Goal, Learned Rules from Past Episodes, Recent Decisions, Area Descriptions.
Output JSON fields: Preflections, Index, Reasoning_Type, Reason, Followed_Rule_Indices
(optional).
Template: {"Preflections":[{"Index":"1","Expected_New_Observations":"...","Potential_
Failure_Mode":"..."}], "Index":"1", "Reasoning_Type":"Room_Function", "Reason":"...",
"Followed_Rule_Indices":[1]}
```

G Additional Qualitative Comparison

Figure 6 compares RSI-Nav with ASCENT in three MP3D episodes targeting a *Potted Plant*, *Nightstand*, and *Table*. In these selected examples, the green RSI-Nav trajectories contain fewer visible detours than the red ASCENT trajectories. In scene (c), for example, RSI-Nav reaches the shown table destination without traversing the additional rooms visited by ASCENT. These cases illustrate individual navigation choices; aggregate path efficiency is evaluated by SPL in the main paper.

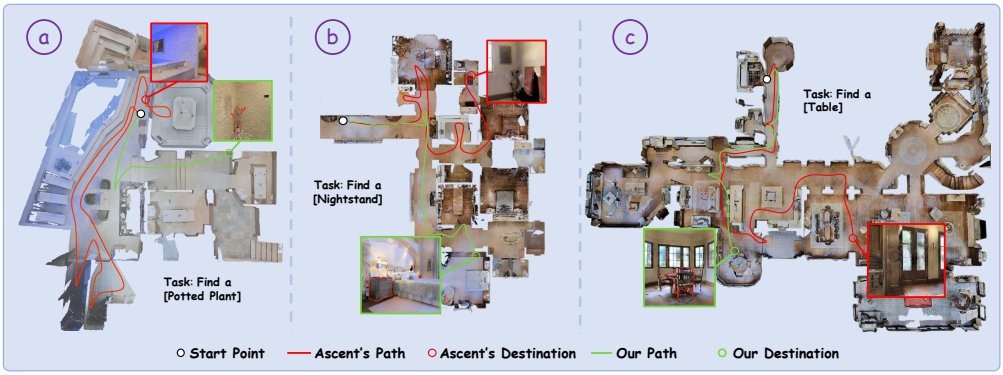

Figure 6: Qualitative trajectories of RSI-Nav and ASCENT (Gong et al., 2026) in three MP3D episodes. Red and green denote the ASCENT and RSI-Nav paths, respectively. These examples illustrate individual behaviors rather than establish statistical significance or global path optimality.