PREACT-NAV: AGENTIC REASONING BEFORE ACTION FOR URBAN NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Urban navigation requires embodied agents to pursue long-horizon goals through local decisions based on egocentric observations. However, existing agentic navigation methods often struggle to translate distant goals into coherent local decisions in large-scale physical environments. Their reliance on linguistic reasoning over transient observations or limited history constrains anticipation of the consequences of actions and future conditions, despite the importance of such foresight for navigating long and complex urban routes. To bridge this gap, we propose PreAct-Nav, an agentic navigation framework that equips frozen policies with anticipatory reasoning for robust urban navigation. Our central idea is to anchor local decisions in persistent medium-horizon subgoals, assess the consequences of predicted actions before execution, and continually update the reasoning context using actual outcomes. At its core, a navigation memory module maintains the active subgoal and relevant experience across decisions, translating distant goals into actionable intermediate objectives. We further introduce a predictive world sandbox that uses an action-conditioned world model (AC-WM) to forecast world dynamics conditioned on candidate movements. A vision-language model (VLM) reasoner interprets these predictions under the current subgoal to retain or revise actions. After execution, real observations are used to assess outcomes, correct inconsistent assumptions, and update memory to continue or reformulate the subgoal. Extensive evaluations demonstrate that the proposed PreAct-Nav improves action selection through memory updates and visual prediction, with more pronounced gains on longer routes and routes with more turns.

1 INTRODUCTION

Urban navigation requires embodied agents to reach a destination described in language through a sequence of local movements (Mei et al., 2026b). The agent sees only part of its surroundings, while the route may extend beyond several intersections and turns. A locally feasible movement is therefore not necessarily useful for reaching the destination. As an example, the route in Figure 1 first requires approaching a crossing before continuing toward the goal. Thus, reliable navigation in such environment requires connecting such intermediate steps to the long-horizon goal while anticipating how local actions may affect subsequent progress.

Recent language and vision-language models (VLMs) have enabled navigation agents to reason explicitly about instructions, observations, and possible actions (Zhou et al., 2024; Long et al., 2024; Ruan et al., 2026). Rather than directly executing a policy prediction, these agents use reasoning to assess where to move next. EvoNav (Dai et al., 2026) further incorporates historical experience and future thoughts by retrieving past experience and predicting subsequent actions and landmarks. These advances establish reasoning over experience as a practical way to improve navigation decisions.

Despite this progress, connecting local reasoning to long-horizon navigation remains challenging in two respects. **First**, historical context records past observations and actions, but provides limited guidance for subsequent movements toward the long-horizon goal. Local decisions may therefore be reasonable in isolation yet inconsistent across steps. **Second**, reasoning about future actions and landmarks in language does not provide visual evidence of what a proposed movement might reveal. A passage that appears plausible from the current view may remain occluded or fail to lead toward

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.

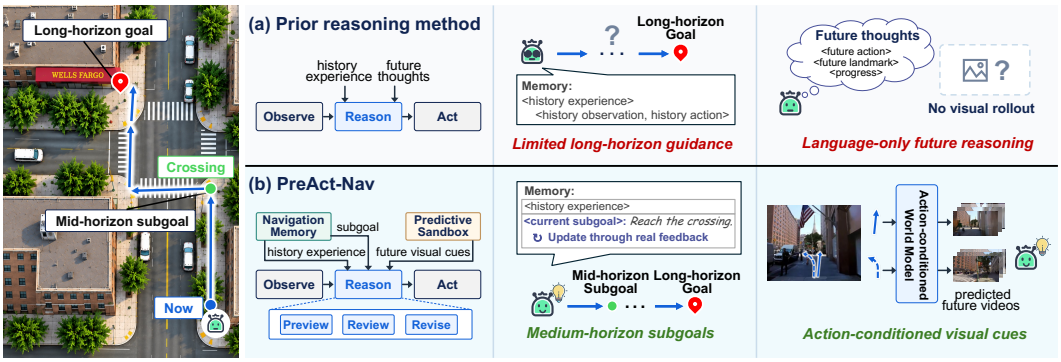


Figure 1: **Reasoning before action with PreAct-Nav.** Navigating across an intersection requires local decisions to serve a long-horizon goal. Compared with reasoning from observed history and language-based future thoughts, PreAct-Nav maintains medium-horizon subgoals to decompose long-horizon navigation tasks and uses action-conditioned future visual cues to preview, review, and revise local actions before execution. Real feedback updates memory for subsequent decisions.

the intended route. These limitations make it difficult to determine whether a locally plausible action will support progress toward the destination.

In this paper, we propose **PreAct-Nav**, a training-free agentic reasoning harness that equips frozen navigation policies with anticipatory reasoning. Our central idea is to maintain medium-horizon subgoals to guide local decisions, evaluate proposed actions using predicted visual outcomes before execution, and update navigation memory using execution feedback. A shared VLM reasoner uses the current subgoal, relevant experience, and visual predictions to decide whether to retain or revise the proposed action. This reasoning loop connects action evaluation and revision before execution with memory updates afterward, without updating model weights.

Specifically, the PreAct-Nav harness organizes a VLM reasoner and two callable tools into a four-stage reasoning loop (Figure 2). During **Action Preview**, the reasoner reads *Navigation Memory*, which maintains the current subgoal and relevant experience across decisions. If the available evidence is insufficient, it queries the *Predictive Sandbox*, which uses an action-conditioned world model (AC-WM) to generate future views for the proposed movement (Bar et al., 2025). During **Action Review**, the reasoner uses predicted visual cues to assess whether the proposed action makes progress toward the current subgoal. It retains a supported action or revises a contradicted one; a revised action returns to preview and review within a fixed reasoning budget. In **Action Execute**, the agent follows the accepted movement. Finally, **Memory Update** compares the expected cues with the next real observation, records the outcome, and updates subgoal progress for the next decision. By coordinating context, tool use, and execution feedback, the harness maintains medium-horizon guidance across local actions: prediction guides revision, and real feedback updates memory.

We evaluate PreAct-Nav on UrbanNav with three navigation policies and two VLM reasoners. Our method achieves relative improvements of 6.71% in success rate (SR) and 4.19% in success weighted by path length (SPL) over the benchmark's original policy. Experiments with other policies and reasoners further assess its applicability beyond a single backbone. Component ablations and analyses of individual decisions examine the roles of memory and prediction, with larger prediction gains on longer routes and routes with more turns. In summary, our contributions are:

❶ We introduce PreAct-Nav, a training-free navigation harness that reviews and revises policy actions using memory and visual prediction before execution.

❷ We develop Navigation Memory to maintain medium-horizon subgoals alongside observed experience and update their progress through execution feedback.

❸ We expose an action-conditioned world model as a Predictive Sandbox, providing future visual cues for action reasoning under the current subgoal.

❹ We evaluate the harness across navigation policies and VLM reasoners, with component ablations and analyses of memory, prediction, and navigation difficulty.

2 Related Work

Navigation reasoning. Vision-and-language navigation grounds instructions in observed scenes, from navigation graphs to continuous environments (Anderson et al., 2018b; Krantz et al., 2020). Learned policies translate observations and goals into local movements (Mei et al., 2026b; Sridhar et al., 2024; Hirose et al., 2026), while CityWalker learns urban navigation from web videos (Liu et al., 2025). LM-Nav composes pretrained models without fine-tuning (Shah et al., 2023), while CompassNav supervises feasible-action decisions (Li et al., 2026). Explicit reasoning supports instruction interpretation (Zhou et al., 2024; Qiao et al., 2025), multi-expert assessment (Long et al., 2024), and retrieved experience with future thoughts (Dai et al., 2026). ReAct and Inner Monologue interleave reasoning with feedback (Yao et al., 2023; Huang et al., 2023), while Toolformer learns when and how to call tools (Schick et al., 2023). Our frozen reasoner coordinates memory and prediction through a harness to review and revise movements before execution.

Agent memory. Memory enables agents to reuse experience without encoding every interaction in model weights. Reflexion stores verbal reflections (Shinn et al., 2023), REMem retrieves temporally grounded events (Shu et al., 2026), and M3-Agent builds multimodal episodic and semantic memories (Long et al., 2026). Agentic Context Engineering incrementally updates contextual playbooks using execution feedback (Zhang et al., 2026). In navigation, DUET combines a growing topological map with local observations (Chen et al., 2022), and PanoNav retains exploration history to avoid repeated local deadlocks (Jin et al., 2026). Our Navigation Memory couples observed experience with an active medium-horizon subgoal that provides shared context for action review and revision. Execution feedback updates its objective and progress cues, so memory tracks both past outcomes and the current purpose of navigation.

Action-conditioned world models. Video generation provides a visual representation of future plans: UniPi generates goal-conditioned videos and extracts control actions from them (Du et al., 2023). Action-conditioned world models instead predict observations for a specified movement. Navigation World Models use such predictions to plan paths or rank proposed trajectories (Bar et al., 2025), while Cosmos Policy fine-tunes video models for visuomotor control and planning (Kim et al., 2026). In computer-use agents, R-WoM grounds language-model simulations in retrieved tutorials (Mei et al., 2026a). Access to a world model, however, does not ensure effective use of its predictions (Qian et al., 2026). Our focus is how visual prediction informs navigation reasoning. The Predictive Sandbox exposes the world model as a callable tool; the VLM interprets predicted visual cues against the current subgoal to retain or revise a movement. Real observations subsequently establish progress and update memory, linking predictive reasoning to execution feedback.

3 Methodology

PreAct-Nav is a training-free agentic reasoning harness for a frozen navigation policy. It organizes a shared VLM reasoner $\mathcal{R}$ and two callable tools into a loop that previews, reviews, and revises actions before execution. **Navigation Memory** maintains a medium-horizon subgoal and observed experience; the **Predictive Sandbox** provides action-conditioned future visual cues. Real execution feedback updates memory for subsequent decisions. Figure 2 summarizes the harness, implemented through structured prompts and tool calls with all model weights frozen.

3.1 Problem formulation

Navigation task and policy actions. Given a language description g of the destination view, the agent must reach the destination and recognize arrival from egocentric RGB observations o_t. At step t, a movement command u_t produces o_{t+1}, whereas STOP ends navigation. The frozen policy returns an action and stopping score, $(a_t^0, p_t^{\text{stop}}) = \pi(g, \mathcal{I}_t^\pi, \mathbf{p}_t)$, using visual inputs $\mathcal{I}_t^\pi$ and, when required, position and orientation inputs $\mathbf{p}_t$. Across policies, actions share a waypoint representation $a_t^0 = (\mathbf{w}_{t,1}^0, \ldots, \mathbf{w}_{t,K}^0)$, where $\mathbf{w}_{t,k}^0 \in \mathbb{R}^2$ is measured in metres relative to the current position and viewing direction: x points right and y forward.

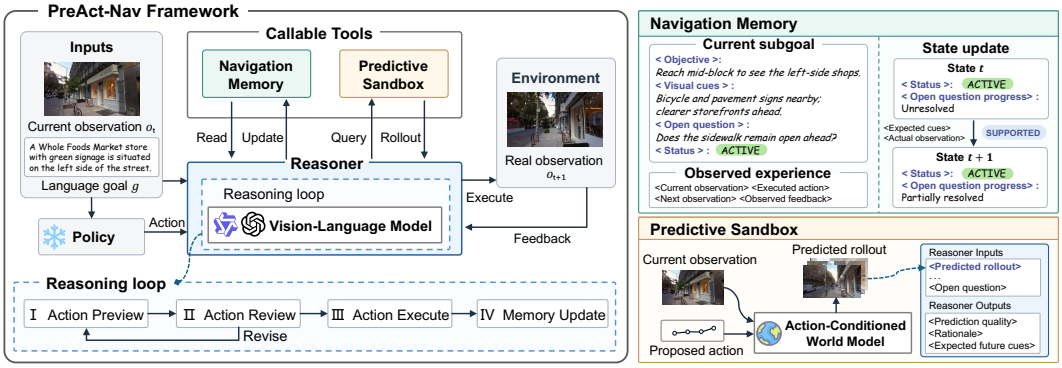


Figure 2: **Overview of PreAct-Nav.** A shared VLM coordinates Navigation Memory and the Predictive Sandbox through Action Preview, Action Review, Action Execute, and Memory Update. Memory provides observed experience and the current subgoal; the sandbox supplies predicted visual cues. Revised actions return to preview. After execution, real observations are checked against expected cues to update subgoal status and open-question progress.

Reasoning context. The reasoner receives the destination, current and selected recent observations $\mathcal{I}_t$, navigation memory $\mathcal{M}_t$, and the policy action:

$$\mathcal{C}_t = (g, \mathcal{I}_t, \mathcal{M}_t, a_t^0), \qquad \mathcal{M}_t = (E_t, S_t), \tag{1}$$

where E_t stores observed experience and S_t represents the current subgoal. Superscript $j = 0$ denotes the policy action and $j > 0$ its successive revisions. The reasoning history $\mathcal{T}_t^j = ((a_t^i, r_t^i))_{i<j}$ retains earlier actions and reviews, with $\mathcal{T}_t^0 = \varnothing$. Observations and memory remain fixed until execution advances t. Subscripts on $\mathcal{R}$ denote role-specific prompts for the same VLM.

3.2 Navigation Memory

Navigation Memory maintains medium-horizon subgoals to decompose long-horizon tasks. Experience E_t records past outcomes; the current subgoal S_t guides action review and revision.

Observed experience. Each record $e_i = (o_i, a_i, o_{i+1}, v_{i+1})$ contains *Current observation*, *Executed action*, *Next observation*, and *Observed feedback*, as shown in Figure 2. Recent images and concise summaries are retained in temporal order to identify actions that failed to produce expected progress. Past actions use the local coordinate frame of their associated observations.

Current subgoal. A subgoal specifies an intermediate objective across local actions:

$$S_t = (z_t, c_t, q_t, \sigma_t), \tag{2}$$

with four fields: *Objective* z_t states what to accomplish; *Visual cues* c_t identify nearby visual anchors and signs of progress or completion; *Open question* q_t records an unresolved route question; and *Status* σ_t indicates whether the subgoal is ACTIVE, ACHIEVED, or CONTRADICTED. The reasoner establishes the subgoal from the destination, current observations, and retained experience.

For example, Figure 2 specifies reaching mid-block to see the left-side shops, with clearer storefronts indicating progress and sidewalk continuity remaining an open question. The subgoal conditions the reasoner; the policy remains conditioned on the final destination g.

State update from execution feedback. After executing a_t, the reasoner compares the next real observation with the *Expected future cues* ν_t retained from action review:

$$v_{t+1} = \mathcal{R}_{\text{update}}(g, S_t, a_t, \nu_t, o_t, o_{t+1}). \tag{3}$$

Feedback v_{t+1} reports SUPPORTED, ACHIEVED, CONTRADICTED, or UNRESOLVED, with a brief explanation of open-question progress. Support indicates compatible progress, achievement requires observed completion cues, contradiction calls for reconsideration, and unresolved feedback preserves the open question. Supported progress can leave a subgoal active with a partially resolved open question (Figure 2).

The observed transition is appended to experience, and the feedback updates the subgoal:

$$E_{t+1} = \mathrm{Select}\big(E_t \parallel (e_t)\big),$$
$$S_{t+1} = \mathrm{Update}(S_t, v_{t+1}), \qquad \mathcal{M}_{t+1} = (E_{t+1}, S_{t+1}). \tag{4}$$

Here $\parallel$ appends a record and Select retains recent experience. Supported or unresolved feedback preserves the active objective; achievement or contradiction closes it before a new objective is established. Only real transitions enter observed experience. Retention settings appear in Appendix D.

3.3 Predictive Sandbox

The Predictive Sandbox exposes an action-conditioned world model $\mathcal{W}$ as a callable tool. From the current observation and proposed action, it generates a short video, or *predicted rollout*, for review under the current subgoal.

Action-conditioned prediction. Let $\mathbf{C}(a_t^j)$ encode the proposed movement as the positions and viewing directions required by the world model. The tool query and returned rollout are

$$Q_t^j = (o_t, \mathbf{C}(a_t^j), \ell, \epsilon), \qquad \widehat{O}_t^j = \mathcal{W}(Q_t^j) = (\widehat{o}_{t,1}^j, \ldots, \widehat{o}_{t,L}^j), \tag{5}$$

where ℓ is a scene-continuation prompt, ϵ controls generation randomness, and L counts predicted frames. Across revisions, the initial observation, prompt, and seed remain fixed; only the proposed movement changes. Appendix D specifies action encoding and generation settings.

Visual evidence collection. The reasoner inspects the rollout and its action using the destination, observations, subgoal, and open question. It returns three fields (Figure 2): *Prediction quality* assesses coherence, movement consistency, and relevance; *Rationale* links visual cues to the action assessment; and *Expected future cues* specify changes to check after execution.

The reasoner looks for progress toward the subgoal, not completion within a single rollout. Incoherent or inconclusive predictions defer to observed evidence. Tool records retain the conditioning action, rollout, and assessment for revision within the current decision.

3.4 Agentic Reasoning Harness

The harness organizes context, tool use, and execution feedback around the VLM's judgments and revisions. Its four-stage loop connects memory retrieval and on-demand prediction to action review, execution, and memory update. Each decision permits at most B reviewed actions, including the policy action, and $B-1$ revisions. The reasoning budget $B \geq 1$ differs from the waypoint count K.

Stage I: Action Preview. Starting with a_t^0, the reasoner performs an observation-based review $\bar{r}_t^j = \mathcal{R}_{\mathrm{review}}(\mathcal{C}_t, \mathcal{T}_t^j, a_t^j, \varnothing)$ using the interface defined below. It requests a preview when the action assessment is unresolved and a forecast could supply missing evidence:

$$\lambda_t^j = \mathbf{1}\Big[\bar{\omega}_t^j = \textsc{Unresolved} \wedge \bar{\kappa}_t^j = 1\Big], \tag{6}$$

where $\bar{\omega}_t^j$ is the observation-based action assessment and $\bar{\kappa}_t^j$ indicates whether a preview would help resolve the open question. The available prediction is

$$\widehat{O}_t^j = \begin{cases} \mathcal{W}(Q_t^j), & \lambda_t^j = 1, \\ \varnothing, & \lambda_t^j = 0. \end{cases} \tag{7}$$

Each action receives at most one preview; otherwise, the observation-based assessment is retained.

Stage II: Action Review. The reasoner reviews the proposed action against the current subgoal, incorporating the predicted rollout when available:

$$r_t^j = (\omega_t^j, \chi_t^j, \rho_t^j, \nu_t^j, \kappa_t^j) = \mathcal{R}_{\mathrm{review}}(\mathcal{C}_t, \mathcal{T}_t^j, a_t^j, \widehat{O}_t^j). \tag{8}$$

The action assessment ω_t^j is Supported, Contradicted, or Unresolved. The remaining entries are prediction quality χ_t^j, rationale ρ_t^j, expected future cues ν_t^j, and preview usefulness κ_t^j. Without a rollout, prediction quality is not assessed; other fields use observations and memory.

A supported assessment accepts the action. A contradicted assessment identifies a reason to change it, such as a movement toward a blocked passage. If the reasoning budget permits, the same VLM generates a revised action with its rationale and expected cues:

$$(a_t^{j+1}, \rho_t^{j+1}, \nu_t^{j+1}) = \mathcal{R}_{\text{revise}}(\mathcal{C}_t, \mathcal{T}_t^j, a_t^j, r_t^j, \widehat{O}_t^j), \quad \text{if } \omega_t^j = \textsc{Contradicted} \wedge j+1 < B. \tag{9}$$

The revision addresses the identified problem under the subgoal and available evidence. The harness appends (a_t^j, r_t^j) to reasoning history. A valid revision must satisfy the action-format and movement checks in Appendix D and differ from previous proposals. Writing validity as $V_t^{j+1} = 1$, with $V_t^0 = 1$, a valid action returns to Stage I at $j \leftarrow j+1$.

Stage III: Action Execute. The agent executes the first supported action. The inner loop also terminates when the budget is exhausted, a revision is invalid or repeated, or the available evidence leaves the assessment unresolved. If no action is supported, it retains the policy action. Let $n_t \le B$ be the number of reviewed actions and $\mathcal{J}_t = \{j : 0 \le j < n_t,\ V_t^j = 1,\ \omega_t^j = \textsc{Supported}\}$. The executed movement is

$$j_t = \begin{cases} \min \mathcal{J}_t, & \mathcal{J}_t \neq \varnothing, \\ 0, & \text{otherwise}, \end{cases} \qquad a_t = a_t^{j_t}. \tag{10}$$

The selected action's assessment and expected cues are retained for memory update. Without intervention, the agent follows the policy action.

Stage IV: Memory Update. The reasoner compares the next real observation with the executed action's expected cues using Equation 3. Equation 4 updates experience and subgoal progress, yielding $\mathcal{M}_{t+1}$ for context $\mathcal{C}_{t+1}$. Prediction guides revision; real feedback establishes progress.

4 Experiments

We evaluate PreAct-Nav on UrbanNav (Mei et al., 2026b) and CityWalker (Liu et al., 2025), examine transfer across frozen policies, and ablate Navigation Memory and the Predictive Sandbox. Qualitative examples illustrate action revision and memory update. Appendices F and G report route-level analyses, reasoning diagnostics, and inference costs.

4.1 Experimental Setup

Tasks and evaluation. UrbanNav (Mei et al., 2026b) provides first-person urban trajectories and language descriptions of destination views. We evaluate on its test-seen and test-unseen splits, with 4,179 and 4,634 episodes after filtering watermark-related descriptions. On CityWalker (Liu et al., 2025), we use 154 routes drawn from nine recordings, with each destination specified by its position relative to the agent. Both tasks use a three-metre success radius. UrbanNav allows 25 decisions per episode and CityWalker allows 100. Methods share the evaluation protocol within each task; Appendix D gives its action and observation rules.

Evaluation metrics. We report success rate (SR), success weighted by path length (SPL) (Anderson et al., 2018a), navigation error (NE), normalized dynamic time warping (nDTW), success weighted by nDTW (SDTW) (Ilharco et al., 2019), and oracle success rate (OSR). SR and SPL measure task completion and path efficiency; the remaining metrics measure final goal distance, reference-path alignment, and whether the goal region is reached at any step. NE is in metres and all other metrics are percentages. Improvements are percentage-point differences computed before rounding. Metric definitions are in Appendix E.

Models and baselines. The default configuration combines the UrbanNav policy (Mei et al., 2026b), Qwen3.8-27B (Qwen Team, 2026), and the official Cosmos3-Nano checkpoint (NVIDIA, 2026) as the action-conditioned world model. We also evaluate GPT-5.6 Sol (OpenAI, 2026) as the reasoner and NoMaD (Sridhar et al., 2024) and OmniVLA (Hirose et al., 2026) as navigation policies. Baselines include the policy alone, a Direct VLM, and NavGPT (Zhou et al., 2024), Open-Nav (Qiao et al., 2025), DiscussNav (Long et al., 2024), and EvoNav (Dai et al., 2026). All reasoning baselines are evaluated with Qwen3.8-27B; NavGPT and Open-Nav are additionally evaluated with GPT-5.6 Sol.

Table 1: **Comparison on UrbanNav.** All methods use the UrbanNav policy, with rows grouped by VLM reasoner. NE is in metres; other metrics are percentages. Best results are bold.

	Test-seen						Test-unseen					
Method	**SR**↑	**SPL**↑	NE↓	nDTW↑	SDTW↑	OSR↑	**SR**↑	**SPL**↑	NE↓	nDTW↑	SDTW↑	OSR↑
UrbanNav	82.79	81.16	3.3596	94.10	81.73	91.91	78.14	74.73	5.9659	87.73	76.13	84.92
Reasoner: Qwen3.8-27B												
Direct VLM	83.13	80.41	3.3192	94.31	82.02	92.49	79.05	75.58	5.9187	88.02	76.99	85.63
NavGPT	83.23	80.54	3.3090	94.35	82.12	92.49	79.11	75.65	5.9017	88.09	77.06	85.67
Open-Nav	83.15	80.45	3.3175	94.31	82.04	92.44	79.24	75.77	5.8964	88.11	77.18	85.74
DiscussNav	83.08	80.47	3.3311	94.27	81.98	92.39	78.90	75.00	5.8659	88.19	76.94	85.69
EvoNav	83.25	80.61	3.2892	94.38	82.16	92.51	78.87	74.59	5.8376	88.16	76.91	85.58
Ours	85.09	82.79	2.8690	96.27	84.21	96.51	83.38	77.87	5.3353	90.47	81.30	90.12
Reasoner: GPT-5.6 Sol												
NavGPT	83.32	81.00	3.2878	94.38	82.23	92.49	78.55	74.56	5.8943	88.00	76.60	85.30
Open-Nav	83.13	80.67	3.3143	94.29	82.02	92.41	78.85	74.70	5.8785	88.19	76.89	85.65
Ours	**85.91**	**84.19**	**2.8075**	**96.46**	**85.07**	**96.77**	**84.18**	**80.57**	**5.2093**	**90.90**	**82.26**	**90.33**

Table 2: **Comparison on CityWalker.** All methods use the frozen CityWalker policy and Qwen reasoner. NE is in metres; other metrics are percentages. Baseline adaptations and evaluation settings are detailed in Appendix D.5.

Method	**SR**↑	**SPL**↑	NE↓	nDTW↑	SDTW↑	OSR↑
Direct VLM	43.51	43.11	18.28	37.44	32.33	43.51
NavGPT	73.38	70.19	10.12	51.00	48.32	73.38
Open-Nav	64.29	60.98	12.21	45.08	41.41	64.29
EvoNav	63.64	61.05	11.97	46.45	42.34	63.64
Ours	**79.22**	**78.94**	**8.31**	**68.77**	**65.85**	**79.22**

The reasoning baselines receive three observed RGB frames, the policy proposal, and policy-centred candidate viewpoints, and are evaluated under the same protocol.

Implementation details. All pretrained model weights remain frozen during evaluation. Navigation policies share the harness interfaces for memory, prediction, and action review. Appendices A, C, and D describe the reasoning loop, prompts, and video generation.

4.2 Main Results

Comparison on UrbanNav. Table 1 compares PreAct-Nav with the original policy and reasoning baselines on test-seen and test-unseen. On test-unseen, Qwen raises SR from 78.14% to 83.38% and SPL from 74.73% to 77.87%, while reducing NE from 5.9659 to 5.3353 m. It also improves nDTW, SDTW, and OSR, indicating better reference-path alignment and more trajectories reaching the goal region. Relative to Open-Nav, the strongest reasoning baseline in test-unseen SR, the gain is 4.14 points. On test-seen, Qwen reaches 85.09% SR and 82.79% SPL, improving over the original policy by 2.30 and 1.63 points. Both splits improve on all six metrics, with a larger SR gain on test-unseen.

Different VLM reasoners. With GPT-5.6 Sol, PreAct-Nav reaches 84.18% SR and 80.57% SPL on test-unseen, and 85.91% SR and 84.19% SPL on test-seen. With the same GPT reasoner, Open-Nav reaches 78.85% SR on test-unseen and NavGPT reaches 83.32% on test-seen; PreAct-Nav exceeds these strongest baselines within the GPT group by 5.33 and 2.58 SR points, respectively. Completion and efficiency improve with both an open-weight and a proprietary VLM.

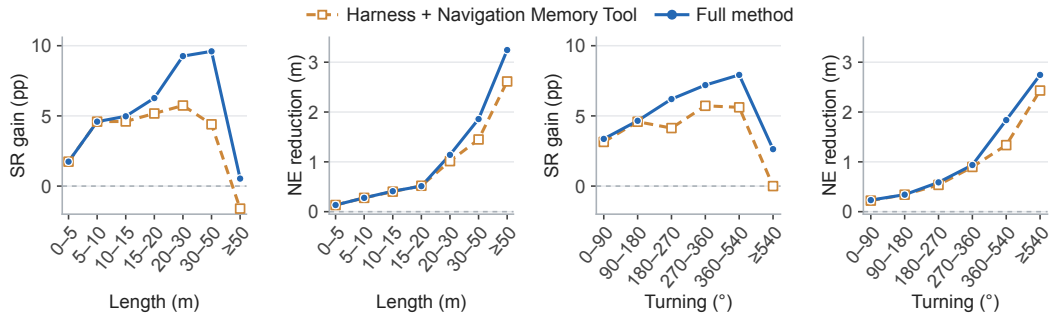
Comparison on CityWalker. Table 2 compares reasoning methods on the frozen CityWalker policy. With Qwen as the shared reasoner, PreAct-Nav achieves 79.22% SR and 78.94% SPL, exceeding NavGPT, the strongest listed baseline in SR, by 5.84 and 8.74 points. It also achieves the lowest NE and highest nDTW, SDTW, and OSR, improving completion, path efficiency, and reference-path alignment. All methods share the CityWalker evaluation protocol.

Table 3: **Comparison across navigation policies on test-unseen.** All pairs use Qwen. Green parentheses show changes from the corresponding frozen policy: percentage points for rate metrics and metres for NE. Changes are computed before rounding.

Policy	SR↑		SPL↑		NE↓		nDTW↑		SDTW↑		OSR↑	
UrbanNav	78.14		74.73		5.9659		87.73		76.13		84.92	
+ harness	83.38	(+5.24)	77.87	(+3.13)	5.3353	(-0.6306)	90.47	(+2.74)	81.30	(+5.17)	90.12	(+5.20)
NoMaD	76.33		69.05		7.2776		82.73		73.55		81.20	
+ harness	79.11	(+2.78)	71.81	(+2.76)	7.2727	(-0.0050)	82.74	(+0.01)	76.13	(+2.58)	81.20	(0.00)
OmniVLA	82.65		78.06		5.8666		87.77		79.98		86.90	
+ harness	86.23	(+3.58)	81.44	(+3.39)	5.7795	(-0.0871)	88.36	(+0.58)	83.33	(+3.35)	88.07	(+1.17)

Table 4: **Component ablation on test-unseen.** ✓/× indicate enabled/disabled components. The first row is the UrbanNav policy. Without either tool, the Reasoning Harness reviews and revises policy actions. NE is in metres; other metrics are percentages.

Reasoning Harness	Navigation Memory Tool	Predictive Sandbox Tool	SR↑	SPL↑	NE↓	nDTW↑	SDTW↑	OSR↑
×	×	×	78.14	74.73	5.9659	87.73	76.13	84.92
✓	×	×	80.56	76.74	5.6410	89.60	78.65	88.37
✓	✓	×	82.37	76.91	5.3986	90.34	80.34	89.90
✓	×	✓	81.48	77.24	5.4791	89.87	79.47	88.89
✓	✓	✓	**83.38**	**77.87**	**5.3353**	**90.47**	**81.30**	**90.12**



(a) SR vs. length (b) NE vs. length (c) SR vs. turning (d) NE vs. turning

Figure 3: **Gains by route length and turning.** SR gain (pp) and NE reduction (m) over the UrbanNav policy, grouped by (a,b) reference-route length and (c,d) cumulative turning. Both variants use the Reasoning Harness and Navigation Memory Tool; the full method also uses the Predictive Sandbox Tool. Positive values indicate improvement.

4.3 Policy Transfer

Table 3 evaluates the harness with three frozen navigation policies on test-unseen. Beyond the UrbanNav policy, NoMaD gains 2.78 SR points and 2.76 SPL points, while OmniVLA gains 3.58 and 3.39 points. SDTW increases from 73.55% to 76.13% for NoMaD and from 79.98% to 83.33% for OmniVLA. OmniVLA improves on all six metrics; for NoMaD, OSR is unchanged and NE and nDTW remain nearly constant. NoMaD's gain primarily reflects better completion and path efficiency. These results show that the harness improves navigation without changing policy weights.

4.4 Ablation Study

Component ablation. Table 4 evaluates the Reasoning Harness and its two callable tools: Navigation Memory Tool and Predictive Sandbox Tool. Without either tool, the harness reviews and revises policy actions, reaching 80.56% SR and 76.74% SPL. Adding the Navigation Memory Tool raises SR to 82.37%, while adding only the Predictive Sandbox Tool yields 81.48%. The full method achieves

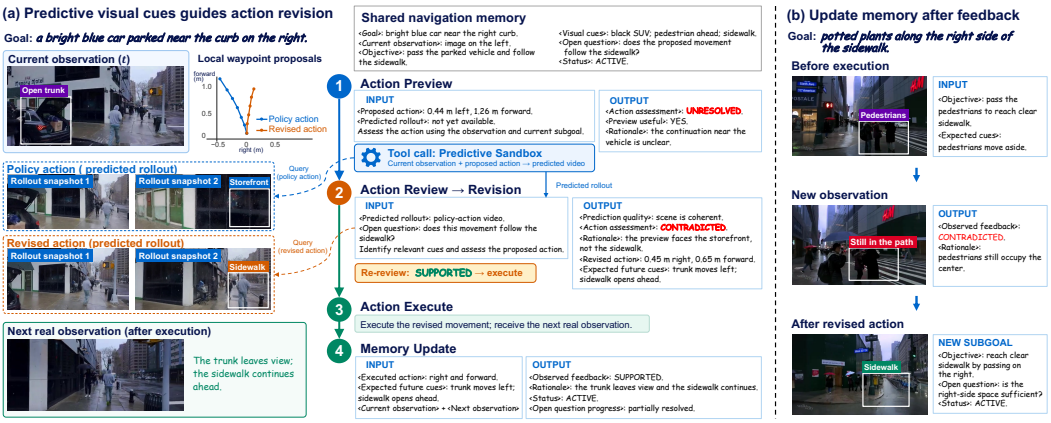


Figure 4: **Action revision and memory update.** (a) Predicted cues prompt a revision from the storefront toward the sidewalk, supported by the next real observation. (b) Pedestrians remain in the passage, prompting a new subgoal to pass on the right.

the highest SR of 83.38%, the lowest NE, and the highest nDTW, SDTW, and OSR. Its SPL is also highest at 77.87%, versus 77.24% with the sandbox alone.

Route length and turning. Figure 3 reports gains over the UrbanNav policy within each route-length and turning interval. The full method improves SR and reduces NE in every interval. Its SR gain reaches 9.60 points on 30–50 m routes and 7.92 points on routes with 360–540 degrees of cumulative turning. The largest NE reductions occur in the longest and most turning-intensive groups, although SR gains decrease in these extreme groups. The Predictive Sandbox Tool generally improves on the harness with the Navigation Memory Tool alone, with larger gains on longer routes and routes with greater cumulative turning. The curves show point estimates; group sizes and the uncertainty protocol are given in the Appendix F.

4.5 Qualitative Results

Figure 4 illustrates how predicted visual cues and real feedback support the reasoning loop. In panel (a), the policy-action preview turns toward a storefront rather than following the sidewalk. This cue motivates a rightward revision, whose preview shows the sidewalk continuing ahead. After execution, the next observation supports the expected passage, and memory records progress under the current subgoal. Panel (b) illustrates the complementary case: pedestrians remain in the passage instead of moving aside, contradicting the expected cues. Memory updates the objective to passing on the right and retains the open question of whether that space is sufficient. The examples connect action preview and review with execution and subsequent memory update.

We also present a qualitative real-robot navigation case in an outdoor plaza (Appendix B). The example illustrates how an exterior-route subgoal connects action review and revision with observation-based memory update.

5 Conclusion and Limitations

We introduced PreAct-Nav, a training-free agentic reasoning harness that reviews and revises navigation actions before execution. Navigation Memory maintains medium-horizon subgoals to decompose long-horizon tasks, while the Predictive Sandbox provides action-conditioned future visual cues. A shared VLM uses this context to retain or revise policy actions, and real feedback updates memory. Experiments across frozen policies show improved navigation success, with greater prediction gains on longer routes and routes with more turns.

The harness remains limited by visual prediction quality and cost. Generated videos can provide ambiguous or inconsistent cues, and video generation dominates inference cost. More reliable, faster predictions are needed to extend this loop to time-sensitive navigation.

AI USE STATEMENT

Generative AI tools and large language models were used during manuscript preparation for drafting and revision, language polishing, grammatical refinement, and reference and LaTeX formatting. Within PreAct-Nav, pretrained vision–language models serve as reasoners for action review, revision, and memory update, while Cosmos provides action-conditioned visual predictions, as described in Sections 3.2–3.4 and Appendix D. UrbanNav and real-world qualitative cases are presented in Section 4.5 and Appendix B, respectively. The authors assume full intellectual responsibility for the methodology, experimental analyses, manuscript content, and associated artifacts.

ETHICS STATEMENT

This work studies navigation using existing urban recordings and does not recruit human participants. Recorded street scenes may contain pedestrians, vehicles, and identifying details; their use and redistribution must respect dataset terms and privacy requirements. Sending observations to externally hosted models requires appropriate authorization and privacy safeguards. Predicted videos may reproduce source-data biases or introduce misleading visual cues. The reported results should not be interpreted as safety guarantees for physical deployment around pedestrians and vehicles.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix A describe the navigation memory, predictive sandbox, and reasoning workflow. Appendix C gives the complete prompt templates, context schemas, response formats, and their correspondence to the method and figures. Section 4.1 and Appendix D document the models, data filtering and partitions, action interfaces, video generation, and evaluation protocol. Appendix E defines the metrics. Section 4.5 and Appendix B present the qualitative cases. Appendices F and G describe supplementary analyses, uncertainty estimation, and inference-cost measurements. The local source package contains prompt templates, figure-generation scripts, figure assets, and result-provenance records; public release of the evaluation runtime remains to be prepared.

REFERENCES

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018a. URL https://arxiv.org/abs/1807.06757.

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 (CVPR)*, 2018b.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15791–15801, June 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 (CVPR)*, pp. 16537–16547, June 2022.

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

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 9156–9172, 2023.

Noriaki Hirose, Catherine Glossop, Dhruv Shah, and Sergey Levine. OmniVLA: An omni-modal vision-language-action model for robot navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

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, 14–18 Dec 2023.

Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. In *NeurIPS Workshop on Visually Grounded Interaction and Language (ViGIL)*, 2019.

Qunchao Jin, Yilin Wu, and Changhao Chen. PanoNav: Mapless zero-shot object navigation with panoramic scene parsing and dynamic memory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(22):18351–18359, 2026. doi: 10.1609/aaai.v40i22.38899.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos Policy: Fine-tuning video models for visuomotor control and planning. In *International Conference on Learning Representations*, 2026.

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

LinFeng Li, Jian Zhao, Yuan Xie, Xin Tan, and Xuelong Li. CompassNav: Steering from path imitation to decision understanding in navigation. In *International Conference on Learning Representations*, 2026.

Xinhao Liu, Jintong Li, Yicheng Jiang, Niranjan Sujay, Zhicheng Yang, Juexiao Zhang, John Abanes, Jing Zhang, and Chen Feng. CityWalker: Learning embodied urban navigation from web-scale videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6875–6885, 2025.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. In *International Conference on Learning Representations*, 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 (ICRA)*, pp. 17380–17387, 2024. doi: 10.1109/ICRA57147.2024.10611565.

Kai Mei, Jiang Guo, Shuaichen Chang, Mingwen Dong, Dongkyu Lee, Xing Niu, and Jiarong Jiang. R-WoM: Retrieval-augmented world model for computer-use agents. In *International Conference on Learning Representations*, 2026a.

Yanghong Mei, Yirong Yang, Longteng Guo, Qunbo Wang, Ming-Ming Yu, Xingjian He, Wenjun Wu, and Jing Liu. UrbanNav: Learning language-guided embodied urban navigation from web-scale human trajectories. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(22): 18505–18513, March 2026b. ISSN 2159-5399. doi: 10.1609/aaai.v40i22.38916.

NVIDIA. Cosmos3-Nano. Hugging Face model card, 2026. URL https://huggingface.co/nvidia/Cosmos3-Nano.

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition, 2026. URL https://openai.com/index/gpt-5-6/.

Cheng Qian, Emre Can Acikgoz, Bingxuan Li, Xiusi Chen, Yuji Zhang, Bingxiang He, Qinyu Luo, Gokhan Tur, Dilek Hakkani-Tür, Yunzhu Li, and Heng Ji. Current agents fail to leverage world model as tool for foresight. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*

(Volume 1: Long Papers), pp. 13686–13723, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.623.

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, 2025. doi: 10.1109/ICRA55743.2025.11127584.

Qwen Team. Qwen3.8-27B. Hugging Face model card, 2026. URL https://huggingface.co/Qwen/Qwen3.8-27B.

Shouwei Ruan, Liyuan Wang, Caixin Kang, Qihui Zhu, Songming Liu, Xingxing Wei, and Hang Su. Brain-inspired spatial intelligence for embodied agents. *Nature Communications*, 17:8061, 2026. doi: 10.1038/s41467-026-74358-5.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551, 2023.

Dhruv Shah, Błażej Osiński, Brian Ichter, and Sergey Levine. LM-Nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 492–504. PMLR, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. doi: 10.52202/075280-0377.

Yiheng Shu, Padmaja Jonnalagedda, Xiang Gao, Bernal Jimenez Gutierrez, Weijian Qi, Kamalika Das, Huan Sun, and Yu Su. REMem: Reasoning with episodic memory in language agent. In *International Conference on Learning Representations*, 2026.

Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. NoMaD: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 63–70, 2024. doi: 10.1109/ICRA57147.2024.10610665.

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

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Y Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In *International Conference on Learning Representations*, pp. 86069–86100, 2026.

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. doi: 10.1609/aaai.v38i7.28597.

SUPPLEMENTARY MATERIAL

The appendix opens with the workflow in Appendix A, followed by the real-world navigation case in Appendix B. Appendix C provides complete prompt templates and their correspondence to the figures; Appendix D documents the implementation and evaluation protocols. Metrics, supplementary analyses, and inference cost appear in Appendices E–G.

A NAVIGATION WORKFLOW

The four-stage navigation loop

Inputs: destination-view description g, local policy π, shared reasoner $\mathcal{R}$, world model $\mathcal{W}$, and operating settings θ.
Initialize observed experience E, subgoal context S, and pending action cues.
For each observation o_t within the navigation budget:
  If an executed decision is pending, compare its expected cues with o_t.
    Record the observed outcome and update $\mathcal{M}_t = (E_t, S_t)$.
  Obtain policy action a_t^0 and STOP score.
  Read memory; establish a subgoal if no active objective remains.
  Initialize the selected movement to the policy proposal.
  Hold observed context and memory fixed within this decision.
  For action index $j = 0, \ldots, B-1$:
    Review the current action using observations and memory.
    If evidence is unresolved and prediction is useful, preview and review it.
    If supported, select this action and exit the inner loop.
    If contradicted and budget remains, construct a valid, distinct revision.
    Continue with that revision; otherwise exit with the policy fallback.
  Retain the movement to execute and its expected visual cues.
  Execute the movement and receive the next observation.
Return: the executed trajectory and terminal outcome.

The procedure summarizes Section 3.4. Environment observations update experience memory, while generated videos remain associated with their prediction calls.

B REAL-WORLD NAVIGATION DEMONSTRATION

Figure 5 presents an online closed-loop run of PreAct-Nav on a wheeled robot navigating toward a yellow food-delivery locker beside a glass entrance. During the run, the harness reviews the policy proposal using the current camera observation and Navigation Memory, querying the Predictive Sandbox when additional visual evidence is needed. The selected movement is executed by the robot, and the resulting camera observation updates memory for the next decision. Predicted rollouts are generated online before action execution, whereas the current and next observations are real camera images acquired during physical navigation. In the illustrated decision, the active subgoal is to follow the exterior walkway until the locker front becomes visible. The policy-action preview approaches the doorway rather than following the exterior route, prompting a rightward revision along the facade. After execution, the next real observation reveals the exterior walkway and locker front. Memory marks the intermediate objective as achieved and establishes the next objective of approaching the locker along the clear paved route. This transition distinguishes subgoal completion from final arrival.

C FULL PROMPT TEMPLATES

This section specifies the prompts for the reasoning loop in Section 3.4: establishing a subgoal, reviewing a proposed action with or without prediction, constructing a revision and updating memory. Templates specify role instructions and response formats; double braces denote runtime substitutions. Line wrapping and JSON indentation are adjusted for readability. Case responses in Figure 4 instantiate these operations for illustration.



Figure 5: **Real-world navigation case.** A robot follows the building exterior toward a delivery locker. The panels connect the current subgoal, visual previews, action revision, and memory update.

C.1 Message construction

Role and visual ordering. Each operation is sent as one user message containing its textual instructions and structured context, followed by real images in chronological order; the last real image is the current observation. These calls do not add a method-specific system message. For predicted review and revision, timestamped predicted frames are appended after the real images and explicitly labeled as predictions. Memory Update receives exactly two real images, before and after execution. The image placeholders below represent visual attachments, not strings sent in place of images.

Shared context. The destination text describes the view at the destination, not directions relative to the current agent. The current action is an 8×2 array in metres, with positive x rightward and positive y forward. Its endpoint description is computed from the last waypoint, not inferred from pixels. Earlier actions retain the coordinate frame in which they were proposed. Within a decision, observations, experience, and the subgoal stay fixed; only the current action and its review history change.

Structured responses. Vertical bars in the printed response formats enumerate valid alternatives; a response returns one value, not the entire list. A predicted review returns the same schema as an observation-based review. Action execution, waypoint decoding, tool dispatch, experience storage, and status transitions are harness operations rather than extra VLM calls.

Table 5: **Runtime substitutions in the prompt templates.** Placeholders are replaced by typed values before serialization; they are not additional model instructions.

Placeholder	Value supplied to the operation
`destination_view`	The unmodified destination-view description g.
`observed_experience`	Recent executed-action records: source step, action in its source frame, measured progress, expected cues, and observed feedback.
`subgoal`	Current objective, visual anchor, completion cue, open question, semantic status, and runtime metadata; `null` before initialization.
`current_action_m`	The proposed waypoint sequence, expressed in the current frame.
`endpoint_x_m`, `endpoint_y_m`	Coordinates of the final waypoint, formatted to three decimal places.
`earlier_reviews`	Ordered action indices, waypoint sequences, and structured reviews; an empty list at the start of a decision.
`action_assessment`	The structured review identifying the contradiction that a revision should address.
`executed_action_m`, `expected_future_cues`	The movement actually executed and the cues retained for checking the next real observation.

The following schema expands the dynamic context. The single records shown stand for variable-length lists; the initial experience and review history are empty. A missing subgoal is serialized as `null`, and `latest_feedback` is absent until a real transition has been assessed. The schema documents data construction, not an additional instruction appended to every request.

Shared context and within-decision review history

```
context = {
  "destination_view": {{destination_view_string}},
  "observed_experience": [
    {
      "source_step": {{source_decision_index}},
      "action_in_source_camera_m": {{executed_waypoints_8_by_2}},
      "progress_m": {{measured_real_progress}},
      "expected": {{expected_cue_string}},
      "observed_feedback": {
        "verdict": {{feedback_enum}},
        "reason": {{observed_evidence_string}}
      }
    }
  ],
  "subgoal": {
    "intent": {{route_intent_string}},
    "anchor": {{visible_anchor_string}},
    "objective": {{intermediate_objective_string}},
    "completion": {{observable_completion_cue_string}},
    "question": {{open_question_string}},
    "status": {{subgoal_status_enum}},
    "created_step": {{creation_decision_index}},
    "progress_m": {{accumulated_real_progress}},
    "latest_feedback": {
      "verdict": {{feedback_enum}},
      "reason": {{observed_evidence_string}}
    }
  }
}
```

Shared context and within-decision review history (continued)

```
earlier_reviews = [
  {
    "action_index": {{inner_loop_index}},
    "action_m": {{proposed_waypoints_8_by_2}},
    "review": {{complete_review_response}}
  }
]
```

C.2 Navigation Memory: establishing a subgoal

The harness issues this request when there is no active subgoal. The prompt receives the destination, retained experience, the previous objective if available, and observed images. The reasoner returns a route intent, visible anchor, intermediate objective, observable completion cue, and open question. The harness initializes `status=ACTIVE`; this status is not generated by the prompt.

Prompt 1. Establishing the current subgoal

```
Establish a tentative medium-horizon route objective for first-person
 navigation. Use a visible anchor and an observable completion cue;
it may require several movements. Do not claim hidden geometry or
arrival. Destination text describes the destination view. Output JSON
 only, each field under 16 words:
{
  "intent": "route intent",
  "anchor": "visible anchor",
  "objective": "intermediate objective",
  "completion": "observable completion cue",
  "question": "unresolved question or NONE"
}.
{
  "destination_view": {{destination_view}},
  "observed_experience": {{observed_experience}},
  "previous_objective": {{subgoal}}
}
```

The compact memory in Equation 2 groups `intent` and `objective` as *Objective*, and `anchor` and `completion` as *Visual cues*. The serialized `question` supplies *Open question*. Creation step and accumulated progress are maintained outside this response.

C.3 Action Preview: observation-based review

The first review examines one proposed movement against real observations and Navigation Memory. It returns both an action assessment and whether a prediction could answer a specific unresolved question. The harness calls the sandbox only for `UNRESOLVED` with `query_useful=true`, as in Equation 6. A visible destination is not required for a local movement to be supported.

Prompt 2. Reviewing the proposed action without a rollout

```
Review the CURRENT proposed movement, not a tournament of candidates.
 Ordered factual images end with the current view. All action
```

Prompt 2. Reviewing the proposed action without a rollout (continued)

```
coordinates are relative to that current camera: x right, y forward,
metres. Destination text describes the destination view. An unseen
destination alone is not a reason to reject local movement. SUPPORTED
 requires visible or credible predicted route support. CONTRADICTED
requires a concrete conflict. Missing evidence alone is UNRESOLVED.
Prediction is useful only for a specific unresolved question about
this movement. Generated views are fallible predictions, not factual
observations or arrival evidence. Reject distorted or action-
inconsistent video cues. No movement occurred between inner
iterations. Current endpoint: right {{endpoint_x_m}} metres, forward
{{endpoint_y_m}} metres.
{
  "context": {
    "destination_view": {{destination_view}},
    "observed_experience": {{observed_experience}},
    "subgoal": {{subgoal}}
  },
  "current_action_m": {{current_action_m}},
  "earlier_reviews": {{earlier_reviews}}
}
No forecast is supplied. A useful request must identify what the
predicted view could clarify.
Output compact JSON only, text fields under 16 words:
{
  "verdict": "SUPPORTED|CONTRADICTED|UNRESOLVED",
  "query_useful": false,
  "question": "specific question or NONE",
  "prediction_quality": "NOT_PROVIDED|COHERENT|INCONCLUSIVE|
INCONSISTENT",
  "reason": "evidence",
  "expected": "cue to inspect after real execution"
}.
```

C.4 Action Review: predicted visual evidence

After a sandbox query, the reasoner receives the same action, observations, memory, and review history, together with the predicted frames. The complete prompt is repeated here to make the two message variants independently usable. Its final instruction states that a forecast of this exact movement is already available, so the same action is not previewed again.

Prompt 3. Reviewing the proposed action with a rollout

```
Review the CURRENT proposed movement, not a tournament of candidates.
 Ordered factual images end with the current view. All action
coordinates are relative to that current camera: x right, y forward,
metres. Destination text describes the destination view. An unseen
destination alone is not a reason to reject local movement. SUPPORTED
 requires visible or credible predicted route support. CONTRADICTED
requires a concrete conflict. Missing evidence alone is UNRESOLVED.
Prediction is useful only for a specific unresolved question about
this movement. Generated views are fallible predictions, not factual
observations or arrival evidence. Reject distorted or action-
```

Prompt 3. Reviewing the proposed action with a rollout (continued)

```
inconsistent video cues. No movement occurred between inner
iterations. Current endpoint: right {{endpoint_x_m}} metres, forward
{{endpoint_y_m}} metres.
{
  "context": {
    "destination_view": {{destination_view}},
    "observed_experience": {{observed_experience}},
    "subgoal": {{subgoal}}
  },
  "current_action_m": {{current_action_m}},
  "earlier_reviews": {{earlier_reviews}}
}
A forecast of this exact movement follows; do not request another
forecast.
Output compact JSON only, text fields under 16 words:
{
  "verdict": "SUPPORTED|CONTRADICTED|UNRESOLVED",
  "query_useful": false,
  "question": "specific question or NONE",
  "prediction_quality": "NOT_PROVIDED|COHERENT|INCONCLUSIVE|
INCONSISTENT",
  "reason": "evidence",
  "expected": "cue to inspect after real execution"
}.
```

Interpreting the review. `verdict` assesses the proposed action, whereas `prediction_quality` assesses the supplied video. A coherent video need not support the action, and an inconclusive video does not itself contradict it. `reason` gives the short, evidence-based rationale; `expected` records the cue to inspect after real execution. With no video, prediction quality is `NOT_PROVIDED`.

C.5 Action Review: constructing a revision

A contradicted review and remaining budget permit a separate revision request. This call receives the current action, its assessment, earlier reviews, and the same observed and predicted images available to the review. It preserves the current subgoal while changing the local movement.

Prompt 4. Revising a contradicted movement

```
Revise the current movement to address its identified contradiction.
Use current observed walkable space; predicted frames can explain the
 problem but do not establish hidden free space. Preserve the medium-
horizon objective. No movement has occurred during this reasoning
loop. Output eight [range_cm,bearing_deg] positions at one-second
intervals, all relative to the CURRENT camera origin. Range is
nonnegative total displacement in centimetres, not a step increment;
bearing -90..90 degrees (negative left, positive right). Choose a
movement distance supported by the currently visible walkable space.
The last position must be at least 10 cm forward. Do not repeat an
earlier action. The new action will be reviewed before execution. If
no grounded revision exists, ABSTAIN.
{
```

Prompt 4. Revising a contradicted movement (continued)

```
  "context": {
    "destination_view": {{destination_view}},
    "observed_experience": {{observed_experience}},
    "subgoal": {{subgoal}}
  },
  "current_action_m": {{current_action_m}},
  "assessment": {{action_assessment}},
  "earlier_reviews": {{earlier_reviews}}
}
Output JSON only, text fields under 16 words:
{
  "decision": "REVISE|ABSTAIN",
  "confidence": "LOW|MEDIUM|HIGH",
  "path": [],
  "grounding": "current visible free space",
  "reason": "how the change addresses the issue",
  "expected": "observable next-view cue"
}.
```

Decoding and re-review. On `REVISE`, `path` must contain eight numeric pairs; the empty array in the printed format is a schema placeholder, not a valid revised path. Each pair specifies range from the current origin in centimetres and bearing in degrees. Appendix D.2 describes the coordinate conversion; Equation 11 specifies the bounds. `ABSTAIN` requires an empty path. The harness checks the decoded action, rejects low-confidence or repeated revisions, and sends a valid new movement back to Action Preview. The review and revision outputs are displayed together in Figure 4, but they come from distinct operations.

C.6 Memory Update: Execution Feedback

After Action Execute, the reasoner compares two real images with the executed movement, current subgoal, and retained expected cues. Predicted frames are not attached to this request. The resulting `verdict` is observed feedback, not another pre-execution action assessment.

Prompt 5. Assessing the observed transition

```
Assess the actual transition against the stored medium-horizon
objective. First image is BEFORE execution, second AFTER. SUPPORTED
means compatible progress, not completion; ACHIEVED requires the
completion cue to be visible. CONTRADICTED requires conflicting
observed evidence; otherwise UNRESOLVED. Historical action
coordinates belong to the BEFORE view. Return JSON
{
  "verdict": "SUPPORTED|ACHIEVED|CONTRADICTED|UNRESOLVED",
  "reason": "specific observed cue, under 20 words"
}.
{
  "destination_view": {{destination_view}},
  "subgoal": {{subgoal}},
  "executed_action_m": {{executed_action_m}},
  "expected": {{expected_future_cues}}
}
```

Table 6: **From observed feedback to memory state.** Feedback is generated by the VLM; storage and semantic-status updates are performed by the harness.

Observed feedback	Subgoal status	Next operation
`SUPPORTED`	`ACTIVE`	Retain the objective and append evidence of compatible progress.
`UNRESOLVED`	`ACTIVE`	Retain the objective and unresolved question.
`ACHIEVED`	`ACHIEVED`	Close the completed subgoal; establish the next objective.
`CONTRADICTED`	`CONTRADICTED`	Reconsider the objective using the conflicting observation.

Every real transition is stored with its feedback. A context-lifetime limit can additionally expire an active subgoal and request a refresh; expiration is runtime bookkeeping, not a claim of semantic failure. The figure's *Open question progress* summarizes the feedback rationale rather than an independently predicted JSON field. Likewise, achieving a subgoal does not authorize stopping at the final destination.

C.7 Predictive Sandbox: world-model request

The world model receives the current image, the proposed movement through the action adapter, and the following fixed scene-continuation text. It does not receive the destination description, reference future, success label, or subgoal as generation text. The subgoal and open question instead guide the VLM's interpretation of the returned video.

Prompt 6. Scene continuation for the action-conditioned world model

```
A continuous realistic first-person video of the same urban sidewalk
scene. Preserve the visible buildings, vehicles, pedestrians,
obstacles, and lighting. The camera motion follows the supplied
camera trajectory.
```

The request uses an empty negative prompt and no added system prompt. Across revisions at the same real observation, the image, scene text, and seed remain fixed while the movement condition changes. Generation settings and action encoding are given in Appendix D.

C.8 Correspondence with the method and figures

Figure 2 uses a compact four-field subgoal; Figure 4 expands the corresponding prompt inputs and outputs. In the parked-vehicle case, Action Preview requests missing evidence, the predicted review identifies the storefront-facing movement, and Revision constructs a sidewalk-following movement for re-review. In the pedestrian case, contradictory real feedback closes the earlier subgoal before a new one is established. The same templates support both transitions; case-specific statements fill the context and response fields rather than change the protocol.

D Implementation Details

D.1 Models and context

The default system uses the UrbanNav navigation policy (Mei et al., 2026b), Qwen3.8-27B (Qwen Team, 2026), and the official pretrained Cosmos3-Nano checkpoint (NVIDIA, 2026). Pretrained models receive no weight updates during evaluation. Structured prompts implement action review, revision, visual comparison, and memory update. The goal description refers to the destination view; spatial phrases retain that reference frame.

Subgoal state. In the compact representation $S_t = (z_t, c_t, q_t, \sigma_t)$, the objective z_t groups the semantic route intent and intermediate objective; cues c_t group the visual anchor and progress or completion evidence. The loop serializes these as `intent`, `objective`, `anchor`, and `completion`;

Table 7: **Prompt outputs in the method and qualitative figures.** Descriptive figure labels expose the semantics of the serialized fields without adding new model outputs.

Displayed field	Runtime field	Role in the method
Objective	`intent`, `objective`	Medium-horizon intent z_t in Equation 2.
Visual cues	`anchor`, `completion`	Observable anchors and progress/completion cues c_t.
Open question	`question`	Unresolved question q_t in memory; review can specify a query about the current action.
Action assessment	review `verdict`	ω_t^j in Equation 8.
Preview useful	`query_useful`	κ_t^j in Equation 8.
Prediction quality	`prediction_quality`	Video assessment χ_t^j.
Rationale	`reason`	Short explanation ρ_t^j, or the explanation accompanying observed feedback.
Expected future cues	`expected`	ν_t^j, retained for the next real transition.
Revised action	decoded `path`	a_t^{j+1}; the figure shows its endpoint rather than all eight waypoints.
Observed feedback	feedback `verdict`	v_{t+1} in Equation 3.
Status	harness-maintained `status`	σ_t; updated according to Table 6.
Open question progress	summary of feedback	A readable description of progress, not a separate response key.
Observed experience	harness transition record	Append $(o_t, a_t, o_{t+1}, v_{t+1})$ after real execution.

archived adapters also use the aliases `subgoal` and `completion_evidence`. The open question and semantic status remain explicit. Creation step, allowed horizon, observed transition count, and accumulated movement are runtime metadata maintained by the harness. Expiration requests a fresh objective without asserting semantic failure. The latest assessment belongs to the associated experience record and may also be cached with the active context. Reasoning indices and rationales remain in the decision log rather than the core memory definition. The final destination remains the input to the local policy.

The preview–review–revise implementation retains four recent experience summaries and refreshes an active subgoal after six decisions if it has not already been achieved or contradicted. Its default inner-loop budget is $B = 2$ reviewed actions.

D.2 Action Representation

UrbanNav produces eight local waypoints. The reasoner receives both their coordinates and a deterministic physical endpoint description, for example a displacement to the right and forward from the current camera. A constructed action uses cumulative ranges and bearings, which an adapter converts to waypoints. Invalid or low-confidence revisions fall back to the policy proposal. Historical action coordinates retain labels identifying their original source frame. NoMaD (Sridhar et al., 2024) and OmniVLA (Hirose et al., 2026) use policy-specific action conversion and operating settings.

For a revised action, waypoint k is specified by displacement r_k from the current origin and bearing β_k relative to forward. The conversion is $x_k = r_k \sin \beta_k$ and $y_k = r_k \cos \beta_k$, with distances in metres and angles in radians. The permitted action set is

$$\mathcal{A}_{\text{rev}} = \{a : 0 \le r_k \le r_{\max}, \quad |\beta_k| \le \beta_{\max}, \quad y_K \ge y_{\min}, \quad k = 1, \dots, K\}. \tag{11}$$

The action interface supplies the range and bearing bounds and the minimum forward displacement. Validation also checks the output format, numerical values, confidence, and whether the revision repeats an earlier proposal. The frozen policy retains its original action interface; the revision prompt specifies its output requirements in Appendix C.

Table 8: **Information exchanged through the harness.** Operations use the same vision–language reasoner. The table lists the inputs and outputs of each prompted operation.

Operation	Principal inputs	Returned information
Subgoal establishment	Destination, observed images, experience, previous objective	Route intent, anchor, objective, completion cue, open question.
Action Preview / Review	Current action, context, earlier reviews, optional predicted rollout	Action assessment, query usefulness, prediction quality, rationale, expected cues.
Action revision	Current action, its review, shared context, available images	Revised path or abstention, confidence, grounding, rationale, expected cues.
Memory Update	Subgoal, executed movement, expected cues, before/after real views	Observed-feedback verdict and rationale.

D.3 Predictive Sandbox

The implementation uses the Diffusers `Cosmos3OmniPipeline` and `CosmosActionCondition` in the `camera_pose` domain. Each query receives the current image, a camera trajectory derived from the proposed movement, and a shared scene-continuation prompt, and returns 60 predicted frames in addition to the input frame. To construct the camera trajectory, we prepend the origin to the eight waypoints and linearly interpolate their positions to 61 poses along the full proposed path. Camera headings follow successive waypoint displacements and are interpolated after angle unwrapping, with the initial heading aligned with the current view. Adjacent poses yield 60 relative transformations, each encoded by three translation coordinates and a six-dimensional rotation representation. These indices parameterize the prediction, not physical execution time. Across revisions, the camera path changes while the image, text, and seed remain fixed. Generation uses seed 0, 30 diffusion steps, guidance 1.0, flow shift 10.0, the 480-resolution tier, ego view, no sound, and no system prompt.

Query records retain the source observation, proposed movement, route question, generated video, and action assessment, linking each predicted rollout to its conditioning action.

D.4 UrbanNav evaluation

The splits contain 4,179 test-seen and 4,634 test-unseen episodes. Each agent receives the destination description and egocentric observations and produces a sequence of local waypoints. All main comparisons use a maximum of 25 navigation decisions per episode and a three-metre success radius. Internal reasoning iterations and tool calls do not count as additional navigation decisions.

D.5 CityWalker evaluation

The additional evaluation uses 154 route windows from nine recordings (IDs 10–18). Goals are fixed route endpoints expressed as rightward and forward displacements in the current frame. The frozen CityWalker policy (Liu et al., 2025) receives five chronological RGB observations and coordinate context. All methods execute at most two metres along the proposed path per decision. Success requires reaching within three metres of the goal. Stationary transitions consume the 100-decision budget. Baselines use task-specific prompt adaptations. Path-length accounting is defined in Appendix E.

E Metrics and Complete Results

We use the standard navigation measures SR, SPL, NE, and OSR (Anderson et al., 2018a;b). For episode i, let S_i denote success, d_i final distance to the goal, l_i reference-route length, and p_i executed waypoint-path length. Let O_i indicate whether any executed waypoint enters the three-metre success radius. For N episodes,

$$\mathrm{SR} = \frac{1}{N}\sum_i S_i, \quad \mathrm{SPL} = \frac{1}{N}\sum_i S_i \frac{l_i}{\max(l_i, p_i)}, \quad \mathrm{NE} = \frac{1}{N}\sum_i d_i, \quad \mathrm{OSR} = \frac{1}{N}\sum_i O_i. \tag{12}$$

Table 9: **Route-group results.** SR/SPL show harness with Navigation Memory Tool / full method from Table 4. Each partition covers all 4,634 episodes. Intervals describe the paired gain in percentage points.

Group	Episodes	SR	ΔSR interval	SPL	ΔSPL interval
Short	3,657	94.26 / 94.56	[0.10, 0.51]	87.65 / 87.91	[0.08, 0.47]
Long	977	37.87 / 41.56	[1.82, 5.57]	36.71 / 40.26	[1.72, 5.39]
Low + medium turn	3,089	94.92 / 95.40	[0.27, 0.72]	88.48 / 88.92	[0.24, 0.66]
High turn	1,545	57.28 / 59.35	[1.04, 3.07]	53.77 / 55.77	[0.98, 2.98]

Trajectory alignment is measured by nDTW and SDTW (Ilharco et al., 2019). For executed trajectory τ_i and reference trajectory τ_i^*,

$$\mathrm{nDTW}_i = \exp\left(-\frac{\mathrm{DTW}(\tau_i, \tau_i^*)}{3|\tau_i^*|}\right), \qquad \mathrm{SDTW} = \frac{1}{N}\sum_i S_i\,\mathrm{nDTW}_i. \tag{13}$$

We average nDTW across episodes. NE is measured in metres; other metrics are multiplied by 100. For SPL, p_i sums the lengths of executed waypoint segments; l_i is the reference-route length. Off-route termination counts as failure before metric aggregation.

E.1 Full-split reporting

Tables 1–4 report all six metrics for the main comparison, policy transfer, and component ablation. Each row comes from a complete evaluation on its split. Table 1 separates test-seen and test-unseen; Tables 3 and 4 use test-unseen.

F Supplementary Analyses

Distribution-based curves. Figure 3 uses reference-length intervals with edges 0, 5, 10, 15, 20, 30, and 50 m, followed by an open-ended interval. Their episode counts are 517, 1,416, 1,086, 638, 540, 250, and 187. The turning intervals have edges 0, 90, 180, 270, 360, and 540 degrees, followed by an open-ended interval, with 922, 1,505, 1,210, 542, 303, and 152 episodes. Intervals include their lower edge and exclude their upper edge; each partition retains all 4,634 episodes. SR gains subtract the original policy's SR, whereas NE reductions subtract the method's NE from the policy's NE. The figure shows point estimates. Paired source-video-cluster bootstrap estimates use the protocol below and are retained with the machine-readable figure data. The subsequent binary-group analysis uses a coarser partition of the same route geometry.

F.1 Route geometry

The reference-length and cumulative-turning partitions are independent. Short routes have reference length ≤ 20 m (3,657 episodes), and long routes have length > 20 m (977). Turning is the sum of absolute heading changes after retaining positions separated by at least 0.5 m. Merging the lower two turning tertiles produces groups of 3,089 and 1,545 episodes at $223.2241°$. All methods use the same groups, determined solely by reference geometry. Table 9 uses the exact episode records for the harness with the Navigation Memory Tool and the full method in Table 4. Aggregating either partition recovers their overall SRs of 82.37% and 83.38%, and SPLs of 76.91% and 77.87%. Bootstrap sampling uses 33 source-video clusters, 10,000 replicates, and seed 20260915; intervals are conditional on the evaluated settings.

F.2 Paired memory and prediction probes

Table 10 summarizes changes to reasoning outputs when memory or predicted-video inputs change. Updating the subgoal context changes 199 of 1,501 policy-action assessments (13.3%) and 134 retain/revise requests (8.9%). In the preview-before-revision probe, revision requests increase from 147 to 214; among the 133 states where both conditions revise, the mean endpoint difference is 0.28

Table 10: **Paired effects on reasoning.** Each comparison holds observed images, destination, and policy proposal fixed. Only the memory update or predicted-video input changes. These are state-level probes, separate from the navigation rollouts.

Changed input	Measurement	Pairs	Result
Updated memory	Policy-action assessment changed	1,501	13.3%
Updated memory	Retain/revise decision changed	1,501	8.9%
Predicted video	States requesting a revision	525	$147 \rightarrow 214$
Predicted video	Endpoint change when both revise	133	0.28 m

Table 11: **Paired reasoning measurements.** Rates are percentages; rate differences are percentage points. The last row is in metres.

Measurement	Pairs	Estimate	95% interval
Memory: changed policy-action assessment	1,501	13.26	$[11.90, 14.70]$
Memory: changed retain/revise request	1,501	8.93	$[7.62, 10.18]$
Revision rate without prediction	525	28.00	$[21.56, 35.44]$
Revision rate with prediction	525	40.76	$[34.63, 47.30]$
Paired revision-rate difference	525	12.76	$[8.90, 16.80]$
Endpoint change when both revise	133	0.278	$[0.215, 0.342]$

m. These measurements describe the reasoner's sensitivity to its inputs, rather than independent accuracy estimates.

At each of 525 fixed states, the construction probe supplies the same destination, observations, memory, and policy action, with or without its predicted rollout. Each condition constructs a new movement independently. These state-level probes are analysed separately from the full-episode navigation results.

The memory analysis includes 1,501 events at which an observed transition produces a new semantic assessment of an active subgoal context. Inherited assessments and expiration alone are excluded. Updates are supported in 633 events, achieved in 283, contradicted in 77, and unresolved in 508. At each event, paired review calls share current images, the policy proposal, observed memory, and recent heading, but receive the previous or updated subgoal context. Achieving an intermediate expectation is distinct from arriving at the final destination.

All probes use frozen Qwen with greedy decoding. Action construction allows 448 output tokens, memory-based review 64, and observed-cue verification 144. Predictions are sampled at 20-frame intervals from existing seed-0 videos exported at 10 fps. All construction, memory-review, and cue-verification calls yield valid structured outputs.

Construction calls revise in both conditions at 133 states and retain the policy in both at 297; prediction introduces 81 revisions and withdraws 14. A predictive cue appears explicitly in 45 construction responses; the remaining 480 cite observed evidence as their basis.

F.3 Observed feedback

Stored action-selection records authorize the alternative at 366 paired states. The complete rollout has 358 matched executions, and 356 of these have an expected observation and a following observed view. The cue-verification prompt receives ordered before/after observed images, without generated frames, reference futures, or success labels. Frozen Qwen judges 183 cues supported (51.4%), 39 contradicted (11.0%), and 134 unresolved (37.6%). These labels are model assessments, not independently annotated ground truth.

G Inference Cost

Experimental setup. We profile the predictive sandbox and the vision–language reasoner of PreAct-Nav separately on a single NVIDIA H200 GPU using BF16 inference. The reasoner is

Table 12: **Predictive sandbox cost on one NVIDIA H200 GPU.** Statistics cover 1,700 rollouts, each containing 60 predicted frames. P90 denotes the 90th percentile.

Mean (s)	Median (s)	P90 (s)	Peak GPU memory (GiB)
11.37	11.35	11.47	34.06

Table 13: **Per-request latency on one NVIDIA H200 GPU at batch size one.** Predicted inputs use frame indices 10, 20, 40, and 60, with index 0 denoting the input observation. Each row times only the named operation; video generation is reported in Table 12.

Operation	Observed / predicted frames	Mean (s)	Median (s)	P90 (s)
Subgoal establishment	3 / 0	5.76	5.21	6.49
Action Preview	3 / 0	5.06	5.04	5.41
Action Review	3 / 4	5.19	5.05	5.65
Action Revision	3 / 0	8.65	9.82	10.77
Memory Update	2 / 0	2.57	2.35	2.91

Qwen3.8-27B with FlashAttention-2, greedy decoding, a batch size of one, and a maximum output length of 384 tokens. Requests use the prompt templates provided in Appendix C, following one warm-up request per operation. Latency includes image loading, preprocessing, model inference, and response decoding, with GPU synchronization at the measurement boundaries. Model initialization and warm-up are excluded.

Predictive sandbox. Cosmos3-Nano generates action-conditioned rollouts with 60 predicted frames, using the 480-resolution setting, 30 denoising steps, and a guidance scale of 1.0. Table 12 summarizes all 1,700 rollouts from the fastest complete generation run. Generation takes 11.37 s on average and uses 34.06 GiB of peak allocated GPU memory; timing includes video export.

Reasoning harness. Table 13 profiles VLM requests within the four-stage loop: subgoal establishment belongs to Navigation Memory, and revision is a separate request within Action Review. Subgoal establishment, Action Preview, and Action Review share the same states, with freshly established subgoals conditioning preview and review. Action Review receives four predicted frames in addition to three observed images. The revision measurement covers observation-based contradicted reviews; Memory Update uses observed before–after transitions. All measured requests, including abstentions, enter the statistics.

The reasoner peaks at 51.89 GiB of allocated GPU memory across measured operations. Action Execute adds no VLM call; prediction and revision are conditional, with revised actions returning to preview and review.