BEYOND THE REMEMBERED WORLD: PREDICTIVE 4D BELIEF FOR PERSISTENT NAVIGATION IN EVOLVING WORLDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Persistent spatial memory enables embodied agents to navigate familiar environments across repeated visits. However, targets may move while unobserved, including during navigation, making remembered locations unreliable by the time an agent arrives. Despite advances in memory retrieval and state prediction, accounting for continued hidden world evolution and revising beliefs under limited visibility remain challenging. We study Evolving-World Navigation, where agents infer target locations from intermittent observations, predict their states at inspection time, and revise beliefs using visual evidence. We propose **EVOLVINGNAV**, which constructs a time-indexed belief from timestamped 3D object histories through a structured persistence–relocation model. The belief distinguishes persistence at the last observed location from relocation to alternative locations and retains probability mass outside the known candidate set. An event-driven filter propagates the current belief as time elapses, forecasts target occupancy at candidate inspection times, and incorporates new RGB-D evidence. Negative observations downweight location hypotheses according to calibrated, visibility-conditioned detection probabilities, while evidence tracking prevents repeated use of the same observations. A frozen, zero-shot vision–language controller uses the updated belief to choose actions and replan. We further introduce **EVOWORLD-BENCH**, a benchmark grounded in human activity traces, comprising 54 scenes and 803,680 tasks with controlled changes before and during navigation. In simulation and real-robot experiments, **EVOLVINGNAV** improves navigation success and search efficiency over the evaluated baselines. Paired experiments show the clearest gains under learnable temporal patterns, while ablations demonstrate the value of preserving uncertainty and incorporating visibility-aware evidence.

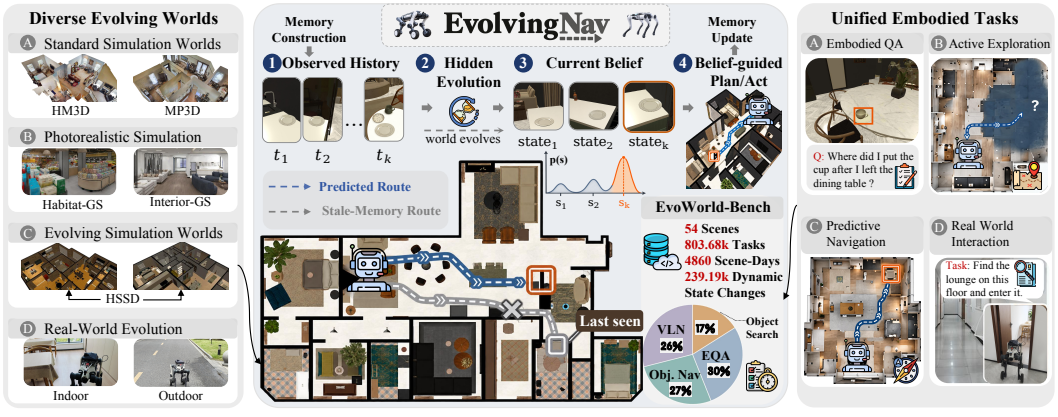

Figure 1: Overview of EVOLVINGNAV for predictive embodied reasoning and action in evolving worlds.

1 INTRODUCTION

Long-lived embodied agents navigate familiar environments across repeated tasks. Spatial maps and object memories preserve observations from earlier visits (Gu et al., 2024; Liu et al., 2024a).

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.

However, those observations may not describe the world when the agent acts. Human activity changes task-relevant states outside observation: a medication box observed on a bedside table may return to a cabinet before the next request. Further changes can occur while the robot travels, invalidating a prediction made at the start of navigation. The agent must therefore plan from observations acquired minutes, hours, or days earlier and estimate what will be true when it inspects a target.

Under partial observability, historical observations are evidence about past states, while the current state remains latent and may evolve. An accurate map can therefore lead an agent to an outdated site, while searching from scratch discards useful history. A time-indexed predictive belief should represent uncertainty over unobserved relocations and support revision during interaction. Candidate destinations have different travel times, so their occupancy probabilities must be evaluated at their respective arrival times. A failed inspection should weaken a location hypothesis only when the site was adequately visible. If changes continue during execution, even an inspected site may later become plausible again.

Existing work has studied both temporal prediction and navigation with moving objects. PredictiveGraphs predicts future object–receptacle states, ranks candidate destinations, and updates its estimates during navigation (Saavedra-Ruiz et al., 2026). Transit-Aware Planning considers portable targets that move while an agent travels (Dorbala et al., 2026). We focus on combining irregular observation histories, candidate-specific arrival times, and visibility-conditioned evidence within one navigation loop. The agent must also retain probability mass outside its known candidate set.

We formalize this problem as **Evolving-World Navigation**. Given a target, a timestamped observation history, candidate locations, and an action budget, the agent must locate and visually verify the target without access to hidden transitions or current ground truth. It receives new observations only along its executed path. Portable-object search provides a concrete instance because people can move objects outside the robot's view. We consider two episode types. In fixed-target episodes, a target may move before the query but remains fixed during the search. In continuing-evolution episodes, it may move while the agent navigates.

We introduce **EVOLVINGNAV** (Figure 1), a navigation agent built around a predictive 4D belief. Its persistent memory records entity identity, 3D spatial context, observation times, and evidence provenance. Using irregularly sampled histories, P4D-NAV predicts whether the target is at its last observed site, at another known site, or somewhere outside the known candidate set. It assigns probability to all three possibilities, keeping alternatives available as the robot gathers new evidence. A frozen, zero-shot vision–language controller uses the belief to call memory, prediction, inspection, exploration, and navigation tools.

The agent closes the loop with an event-driven predict–observe–replan filter. Before choosing a destination, it predicts target occupancy at each candidate's estimated arrival time and weighs that prediction against expected new coverage and travel cost. It then moves in short segments and updates the belief using RGB-D observations gathered along the way and at inspection sites. A clear view of an empty site can redirect its route; an occluded view leaves the corresponding hypothesis largely intact. Time-valid evidence rounds prevent correlated frames from being counted repeatedly. As time passes, the model can also restore probability to a site inspected earlier.

We introduce **EVOWORLD-BENCH** to evaluate these decisions under changing conditions. Grounded in human trajectories, it contains 54 evolving scenes and 803,680 task instances spanning state prediction, object navigation, vision-and-language navigation, and embodied question answering. The benchmark measures how predictions affect first inspection, recovery, path efficiency, and online replanning. Paired static, routine, and random worlds keep scenes and queries fixed while varying temporal structure. These comparisons test when learnable patterns help an agent predict and search.

Across the navigation suite, **EVOLVINGNAV** improves initial inspection and eventual search (Table 1). When targets can move during execution, it recovers more reliably and avoids unnecessary travel (Table 2). Paired experiments show the clearest benefit when changes follow learnable routines. Ablations examine the roles of alternative hypotheses and visibility-aware updates. Held-out scenes, cross-benchmark tests, different frozen VLMs, and physical-robot trials assess performance across the evaluated settings.

Our contributions are threefold:

- **Problem setting:** We study navigation with hidden changes before a query and, in continuing-evolution episodes, during execution. The agent must reason about arrival times and visually verify the target.

- **Belief-driven agent:** We introduce **EVOLVINGNAV**, combining a time-indexed belief over known and unknown locations with an event-driven predict–observe–replan filter.
- **Benchmark and evaluation:** We introduce **EVOWORLD-BENCH** and use paired temporal controls, simulation, and physical-robot trials to evaluate prediction, evidence-based recovery, and navigation efficiency.

2 RELATED WORK

Navigation, memory, and changing worlds. Embodied navigation combines perception, spatial reasoning, and action; open-vocabulary maps and scene graphs ground language in spatial representations (Anderson et al., 2018; Savva et al., 2019; Chaplot et al., 2020; Xin et al., 2026; Huang et al., 2023; Gu et al., 2024; Werby et al., 2024; Li et al., 2026a; Huang et al., 2026). Foundation-model agents combine maps, tools, and navigation skills (Yokoyama et al., 2024; Long et al., 2024; Ziliotto et al., 2025; Li et al., 2026b; Zheng et al., 2022; Ginting et al., 2024; Zhang et al., 2025). Lifelong benchmarks evaluate navigation and memory across tasks (Khanna et al., 2024b; Yadav et al., 2025), while embodied QA benchmarks assess question answering from episodic observations (Majumdar et al., 2024). Dynamic maps and 4D memories support updates and spatiotemporal retrieval (Sohn et al., 2025; Gorlo et al., 2026; Bescos et al., 2018; Narayana et al., 2020; Liu et al., 2024a).

Predictive memory and human routines. Predictive navigation models infer object locations from partial histories using discrete states or continuous spatial representations (Kurenkov et al., 2023; Saavedra-Ruiz et al., 2026; Argenziano et al., 2026). PredictiveGraphs combines future-state prediction, Top-k navigation, online observations, and agent tools; our focus is an arrival-time filtering loop for continued hidden evolution, with time-valid evidence and candidate reopening. HOMER/HOMER+, STREAK, and personalized navigation model household routines or context drift (Patel & Chernova, 2023; Patel et al., 2023; Bartoli et al., 2025; Wang et al., 2026). HD-EPIC and ParaHome provide human–object interaction traces (Perrett et al., 2025; Kim et al., 2025). EVOWORLD-BENCH uses human traces and separately tracked simulated evidence to constrain executable histories; paired worlds test temporal regularity beyond location frequency.

3 PREDICTIVE 4D BELIEF NAVIGATION

EVOLVINGNAV maintains a current-state belief from timestamped 3D histories and RGB-D evidence; P4D-NAV separates persistence from relocation (Figure 2).

3.1 PROBLEM FORMULATION

An environment contains a navigable space $\mathcal{X}$, entities $\mathcal{O}$, and candidate states $\mathcal{C}_o$ with inspection viewpoints. Before query time t_q, the agent has only the causal history

$$\mathcal{H}_{\leq t_q} = \{(I_k, D_k, \xi_k, t_k) \mid t_k \leq t_q\}, \tag{1}$$

where I_k, D_k, ξ_k denote RGB, depth, and camera pose. The hidden state lies in $\widetilde{\mathcal{C}}_o = \mathcal{C}_o \cup \{\text{unknown}\}$; later out-of-view transitions are also hidden.

Given a target, initial pose x_0, memory, and action budget, the agent observes z_j after acting and must stop at a valid target viewpoint. It optimizes

$$\pi^* = \arg\max_{\pi} \ \mathbb{E}_\pi[\mathbb{I}(\text{success}) - \lambda_d L - \lambda_n N_{\text{inspect}}], \tag{2}$$

where L is the travel distance and N_{inspect} is the inspection count.

3.2 CAUSAL 4D MEMORY AND HISTORY ENCODING

The 4D memory registers RGB-D observations in a shared frame, associates entities using semantics, geometry, and time, and materializes versioned deltas causally:

$$\mathcal{M}_{\leq t} = \text{Materialize}(\Delta\mathcal{M}_0, \ldots, \Delta\mathcal{M}_t) = (\mathcal{G}_t, \mathcal{V}_t, \mathcal{B}_t), \tag{3}$$

where $\mathcal{G}_t, \mathcal{V}_t, \mathcal{B}_t$ store relations, time-valid versions, and provenance. Changes append versions without erasing history; perception and versioning details are in Appendix A.2 (Liu et al., 2024b; Ravi et al., 2025; Radford et al., 2021).

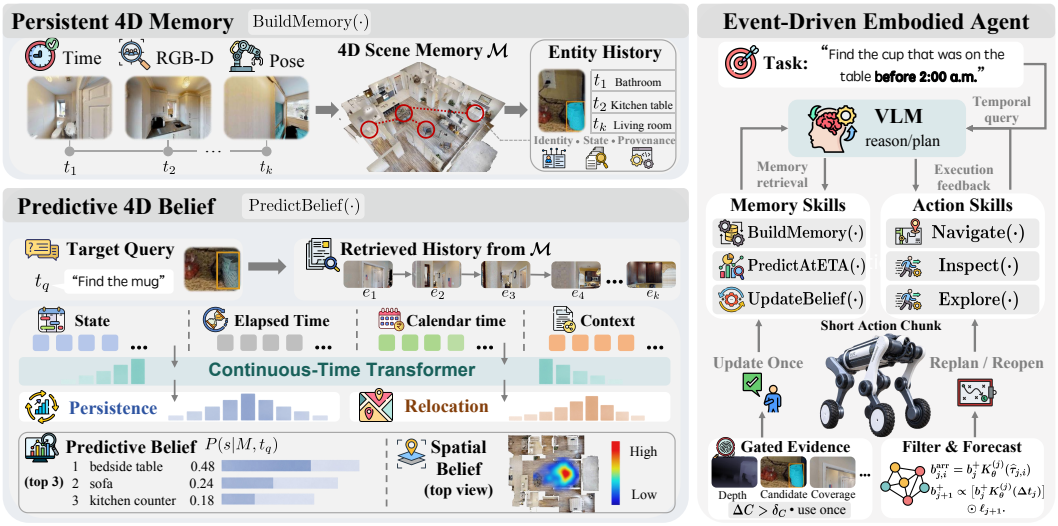

Figure 2: EVOLVINGNAV couples persistent 4D memory, predictive belief, and an embodied agent that updates and replans from stepwise evidence.

Target-conditioned history and continuous time. For target o, a typed query returns the most recent K causal events

$$H_o = \{e_k = (o_k, s_k, t_k, z_k, c_k, v_k)\}_{k=1}^{K}, \tag{4}$$

where o_k, s_k, t_k, z_k encode entity, candidate state, time, and $\{\text{seen}, \text{notseen}\}$ evidence; c_k, v_k encode confidence and visibility. Related context events are causal, candidate encodings capture function and geometry, and `unknown` absorbs unmapped states.

Each event token combines semantic, evidence, and temporal features:

$$\begin{aligned}\mathbf{x}_k = {} & E_o(o_k) + E_s(s_k) + E_z(z_k) + E_c(c_k) + E_v(v_k) \\ & + \phi_\Delta(t_q - t_k) + \phi_{\text{cal}}(t_k),\end{aligned} \tag{5}$$

where ϕ_Δ encodes log elapsed time and ϕ_{cal} encodes hour and weekday. A query token specifies the target, time, and last observed state; a compact Transformer yields

$$(\mathbf{h}_1, \ldots, \mathbf{h}_K, \mathbf{h}_q) = \text{CTTransformer}(\mathbf{x}_1, \ldots, \mathbf{x}_K, \mathbf{x}_q). \tag{6}$$

The encoding captures irregular intervals, evidence, and event order without hidden labels.

3.3 PERSISTENCE–RELOCATION BELIEF

We predict persistence at the last observed state or relocation elsewhere. Episode validity guarantees a last positively observed state s_{last} (Appendix B.4). The persistence branch predicts

$$\rho_q = P(s^o_{t_q} = s_{\text{last}} \mid H_o, t_q) = \sigma(\mathbf{w}_\rho^\top \mathbf{h}_q). \tag{7}$$

For each alternative $i \in \widetilde{\mathcal{C}}_o \setminus \{s_{\text{last}}\}$, a shared pointer head computes

$$a_i = \frac{(W_q \mathbf{h}_q)^\top (W_c \mathbf{g}_i)}{\sqrt{d}} + \psi(\mathbf{h}_q, \mathbf{g}_i), \qquad \pi_q(i) = \frac{\exp a_i}{\sum_{u \in \widetilde{\mathcal{C}}_o \setminus \{s_{\text{last}}\}} \exp a_u}, \tag{8}$$

where $\mathbf{g}_i$ represents candidate i, d is the projection dimension, and ψ measures learned compatibility. The prior is

$$b_q^-(i) = \begin{cases} \rho_q, & i = s_{\text{last}}, \\ (1 - \rho_q)\pi_q(i), & i \neq s_{\text{last}}. \end{cases} \tag{9}$$

We train with $\mathcal{L}_{\text{state}} = -\log b_q^-(s^o_{t_q})$ without change labels; the normalized belief represents returns and alternatives.

3.4 PREDICTIVE EMBODIED AGENT

A frozen VLM invokes memory, prediction, navigation, inspection, and exploration tools. Action chunks, new coverage or evidence, ETA changes, and arrivals trigger decision epochs. The encoder produces $\mathbf{h}_j$ for the row-normalized operator

$$[K_\theta^{(j)}(\Delta t)]_{r,u} = \mathrm{softmax}_u\, f_\theta(\mathbf{h}_j, \mathbf{g}_r, \mathbf{g}_u, \phi_\Delta(\Delta t)). \tag{10}$$

The operator shares the encoders in Equations (6) and (8); a separate head learns chronological state transitions through row-wise cross-entropy (Appendix A.4). Initialized with $b_0^- = b_q^-$, the filter propagates after each chunk:

$$b_{j+1}^- = b_j^+ K_\theta^{(j)}(t_{j+1} - t_j). \tag{11}$$

Thus $b_j^\pm$ denotes the current-time belief, separate from arrival forecasts.

Arrival-aware action selection. For candidate viewpoints $\mathcal{W}_i$, let $d_{j,i} = d_{\mathrm{geo}}(x_j, \mathcal{W}_i)$ and $\widehat{\tau}_{j,i}$ be the geodesic distance and estimated travel-plus-inspection time. The agent selects

$$i_j^* = \arg\max_{i \in \mathcal{C}_o} \frac{p_{j,i}^{\mathrm{arr}}\, \widehat{q}_{j,i}^{\mathrm{new}}}{d_{j,i} + \lambda_t \widehat{\tau}_{j,i} + \lambda_{\mathrm{inspect}}}, \tag{12}$$

where $p_{j,i}^{\mathrm{arr}}$ is defined below and $\widehat{q}_{j,i}^{\mathrm{new}}$ is the detection probability from newly covered surfaces. Exploration expands the candidate set when unknown-state utility is highest.

Visibility-qualified measurement update. A posed RGB-D view can update every covered candidate. A calibrated detector estimates $\widehat{r}_{j,i} = P(\mathrm{detect} \mid s_{t_j} = i, \mathbf{f}_{j,i})$ from projected candidate geometry, online depth, and view features, without access to ground-truth target masks or poses or simulator visibility flags. Given no detection,

$$\ell_j(i) = P(y_j = \varnothing \mid s_{t_j} = i) = \begin{cases} 1 - \widehat{r}_{j,i}, & i \in \mathcal{C}_o, \\ 1, & i = \mathrm{unknown}, \end{cases} \tag{13}$$

and the measurement posterior is

$$b_j^+(i) = \frac{b_j^-(i)\ell_j(i)}{\sum_{u \in \widetilde{\mathcal{C}}_o} b_j^-(u)\ell_j(u)}. \tag{14}$$

An update uses only new surface coverage exceeding $\delta_C = 0.05$, preventing duplicate evidence from overlapping views. Dynamic reopening starts a new time-valid round, allowing the same viewpoint to test a changed state. A verified match ends the search.

Posterior propagation and replanning. The current posterior is forecast to each candidate ETA for action selection, without replacing the current-time filter:

$$b_{j,i}^{\mathrm{arr}} = b_j^+ K_\theta^{(j)}(\widehat{\tau}_{j,i}), \qquad p_{j,i}^{\mathrm{arr}} = b_{j,i}^{\mathrm{arr}}(i). \tag{15}$$

Fixed-target episodes use identity dynamics; continuing evolution follows Equations (11) and (15). At each epoch, the agent propagates belief, incorporates new evidence once, forecasts arrival-time states, and replans (Appendix A.4).

4 EVOWORLD-BENCH: EVALUATING NAVIGATION IN EVOLVING WORLDS

EVOWORLD-BENCH evaluates decisions based on time-ordered histories while the world changes outside observation. Its 54 scenes and 803.68k task instances combine persistent histories, executable placements, and controlled dynamics.

4.1 TASKS AND CONTROLLED DYNAMICS

At the task-modality level in Figure 1, the pool comprises object search (17%), object navigation (27%), vision-and-language navigation (VLN; 26%), and embodied question answering (EQA; 30%) (Majumdar et al., 2024; Sakamoto et al., 2024). These labels describe task goals; the protocols below distinguish prediction, search, evidence, and online dynamics.

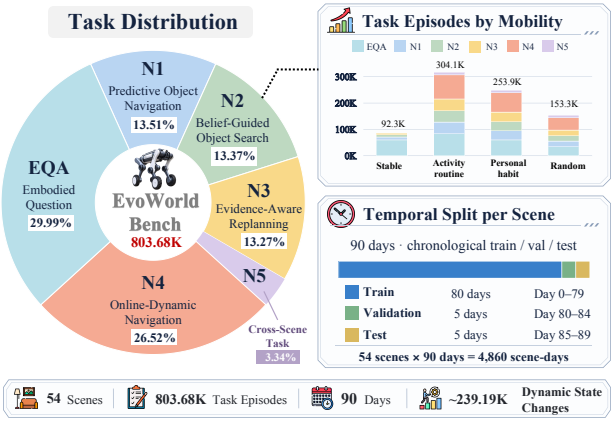

Figure 3: Overview of EvoWorld-Bench evaluation protocols.

The complementary breakdown by evaluation protocol in Figure 3 comprises N1 (13.51%), N2 (13.37%), N3 (13.27%), N4 (26.52%), N5 (3.34%), and EQA (29.99%). In this second view, VLN-conditioned episodes are assigned to their corresponding navigation protocol rather than a separate category. Prediction evaluates the region/receptacle belief independently of control; navigation uses the same target–time queries (Figure 3 and Table 8). N1 inspects the Top-1 destination; N2 uses the full belief for cost-aware search; N3 adds visibility-qualified updates and recovery. N1–N3 fix the target after the query. N4 advances the world clock during execution, permitting further hidden transitions. N5 applies these protocols to held-out scenes and households.

VLN grounds language goals in the same histories; EQA probes current and historical states without future evidence. Our experiments emphasize prediction and navigation, with EQA as a memory diagnostic. Controlled comparisons match public observations, candidates, geometry, starts, controllers, and budgets.

Paired *static*, *routine*, and *random* worlds share public setup variables. Static preserves the last supported state; routine follows household dynamics; random matches legal candidates and marginal movement statistics while removing temporal dependencies. Improvements specific to routine worlds thus support the use of historical regularity over category frequency or spatial convenience.

4.2 World Construction and Past-Only Evaluation

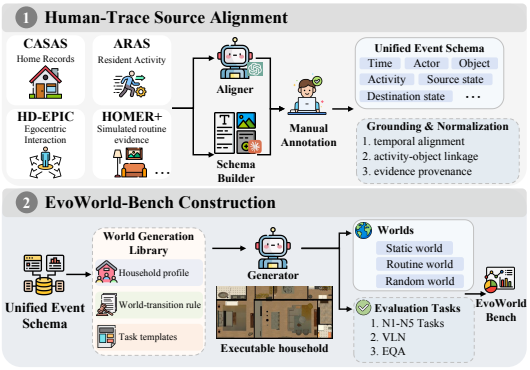

Figure 4: Construction of EvoWorld-Bench.

Human traces constrain generation; source trajectories are not copied (Figure 4). CASAS supplies longitudinal occupancy and timing statistics; ARAS and OPPORTUNITY add activity-context coverage; and HD-EPIC and ParaHome provide object–action and motion evidence (Cook et al., 2013; Perrett et al., 2025; Kim et al., 2025). HOMER+ is tracked separately as simulated long-horizon routine evidence (Patel et al., 2023). Event records preserve source identifiers and distinguish measured, environment-derived, simulated, and benchmark-authored quantities (Appendix B.1).

Across 54 HSSD scenes (Khanna et al., 2024a), household habits and stochastic exceptions generate chronological histories. Stable, routine, personal, and irregular regimes test persistence, activity regularity, owner-specific habits, and uncertainty (Table 7); regime labels remain hidden from the agent. Habitat replay ensures semantically valid, collision-free placements and reachable inspection viewpoints.

Patrols collect visibility-qualified RGB-D observations using only maps, time, travel cost, and observed history. Queries expose only observations available by the query time, the last supported state, elapsed/calendar time, candidates, and spatial context; current states, transitions, activities, mobility labels, and oracle viewpoints remain private. Splits group complete temporal sequences, paired worlds, and near-duplicate query/transition groups. Audits cover temporal and metadata leakage, split overlap, paired-world consistency, physical validity, and patrol access to hidden state. Stratification covers mobility, transition, staleness, spatial difficulty, and generalization.

5 EXPERIMENTS

We test whether predictive 4D memory improves current-state inference, embodied search, and recovery when observations contradict prior beliefs. Controlled EVOWORLD-BENCH and robot comparisons use the same public histories, legal candidates, inspection viewpoints, low-level control, and action budgets. Cross-environment evaluations retain the native policies of released agents and provide system-level generalization evidence. Component ablations support causal comparisons. Further protocol, implementation, and analysis details are summarized in Appendices A–D.

5.1 SETUP

Datasets. Our primary evaluation uses the complete EVOWORLD-BENCH N1–N5 suite for First-Inspection navigation, belief-guided search, evidence-aware replanning, online dynamics, and cross-scene transfer. FindingDory (Yadav et al., 2025) and GOAT-Bench (Khanna et al., 2024b) test long-horizon memory and repeated-goal navigation. We additionally evaluate one native task in each of MP3D, HM3D, Habitat-GS, and InteriorGS, and run 64 matched LYNX M20 trials per method.

Real-world setup. We use the DEEP Robotics LYNX M20 for the primary quantitative evaluation over 64 matched trials, with X30 and Lite3 evaluated on matched 32-block transfer subsets. Platform-specific perception and low-level navigation use a common interface, and all methods receive identical pre-query histories, candidate locations, start conditions, and robot-specific control stacks. Each 360-second episode requires online target confirmation within three inspections.

Controllers and baselines. Our main method uses a frozen GPT-5.6-Luna controller in a zero-shot setting: the VLM receives no navigation-task fine-tuning and invokes the fixed memory, prediction, inspection, exploration, and navigation tools. The prompt, tool interface, and inference budget remain fixed within each comparison. Baselines span navigation, structured memory, and predictive world models; paired runs with GPT-4o, GPT-5.5, GPT-5.6-Luna, and Qwen2.5-VL-3B/32B test controller dependence.

Metrics. We evaluate prediction using Top-1, MRR, NLL, ECE, and true-state rank, and navigation using SR, SPL, First-Inspection SR, Recovery SR, inspection count, distance, and time. Main comparisons report five-seed variation; paired static, routine, and random worlds isolate learnable temporal structure.

5.2 MAIN RESULTS

Table 1: Cross-benchmark comparison (%). ± denotes standard deviation over five seeds.

	FindingDory		GOAT-Bench			EVOWORLD-BENCH		
Method	**HL-SR ↑**	**HL-SPL ↑**	**SR ↑**	**SPL ↑**	**Repeat-SR ↑**	**First-Inspection SR ↑**	**Search SR ↑**	**SPL ↑**
Open-vocabulary and lifelong navigation								
VLFM (Yokoyama et al., 2024)	13.42	10.86	15.79	5.45	11.82	20.65	28.36	24.67
ZSON (Majumdar et al., 2022)	27.89	21.81	29.64	2.45	16.37	26.84	32.13	28.55
FindingDory Agent (Yadav et al., 2025)	**52.44**[PR]	**40.92**	–	–	–	–	–	–
LagMemo GLUE (Zhou et al., 2025)	32.79	23.25	20.61	1.74	11.27	30.18	42.27	33.82
Structured spatio-temporal memory								
HOV-SG (Werby et al., 2024)	21.47	15.20	28.37	2.23	12.38	36.46	52.32	40.61
DynaMem (Liu et al., 2024a)	30.30	22.78	14.54	1.71	15.79	45.33	71.94	56.83
Predictive world and state modeling								
SLaTe-PRO (Patel et al., 2023)	18.18	14.53	9.41	0.99	6.38	13.00	36.33	26.37
SGM+NEP (Kurenkov et al., 2023)	34.20	19.48	10.67	1.64	2.56	27.26	41.82	37.00
FlowMaps (Argenziano et al., 2026)	13.03	10.24	25.07	1.32	7.25	14.33	32.67	17.35
PredictiveGraphs (Saavedra-Ruiz et al., 2026)	24.24	13.50	10.27	1.80	7.48	36.18	43.67	40.36
EVOLVINGNAV (ours)	**53.22** ±3.87	**38.83** ±2.43	**35.43** ±2.16	**12.98** ±0.92	**19.31** ±0.68	**61.32** ±3.235	**86.18** ±2.07	**70.15** ±1.74

[PR] Paper-reported under the cited native protocol; unmarked baseline scores are our reproductions under the method-preserving protocol in Appendix A.7.

Comparison across embodied benchmarks. EVOLVINGNAV performs strongly across FindingDory, GOAT-Bench, and EVOWORLD-BENCH (Table 1). On EVOWORLD-BENCH N1–N5, First-Inspection SR increases from 45.33% (DynaMem) to 61.32%, and Search SR from 71.94% to 86.18%. On FindingDory, it achieves the highest HL-SR (53.22%) among the listed methods and the second-highest HL-SPL (38.83%), behind FindingDory Agent (40.92%). The gains on EVOWORLD-BENCH show the benefit of predictive belief under hidden world evolution; results on FindingDory and GOAT-Bench indicate broader applicability to embodied navigation.



Method	Checks	Path length	Result	SPL
Last Seen	3	6.49 m	Failed	0.000
Nearest	2	8.19 m	Success	0.379
Ours	1	3.10 m	Success	1.000

Figure 5: Predictive, last-seen, and nearest-first search on EVOWORLD-BENCH.

Table 2: N4 online-dynamic navigation with execution-time target motion.

Method	Dynamic SR ↑	Online Recovery SR ↑	Excess Distance (m) ↓	Revisit Success ↑
SLaTe-PRO (Patel et al., 2023)	34.7	21.5	18.9	8.7
SGM+NEP (Kurenkov et al., 2023)	41.3	27.8	15.6	12.4
FlowMaps (Argenziano et al., 2026)	29.6	18.9	22.4	6.5
PredictiveGraphs (Saavedra-Ruiz et al., 2026)	46.8	36.7	13.2	18.9
EVOLVINGNAV (ours)	**65.2**	**58.7**	**8.4**	**32.6**

Online navigation under execution-time dynamics. Figure 5 shows how a routine-conditioned belief avoids an obsolete last-seen location. To isolate the dynamic closed loop from episodes in which motion is permitted but no transition is scheduled, Table 2 reports the same preselected N4 episodes for every method, with a target transition scheduled inside a common, fixed post-query window.

Relative to PredictiveGraphs, EVOLVINGNAV improves Dynamic SR by 18.4 points and Online Recovery SR by 22.0 points, while reducing excess distance by 4.8 m and increasing successful revisits by 13.7 points. These results support the execution-time contribution: current-time propagation and time-valid evidence rounds let the agent recover after a target moves, while candidate reopening turns revisits into useful actions rather than permanent exclusions.

Generalization and physical deployment. Across five benchmark tasks, EVOLVINGNAV leads on seven of ten metrics (Table 9). Across 64 matched M20 trials, the agent improves both success and efficiency (Table 3); details are in Appendix D.

Table 3: Primary LYNX M20 search results.

Method	First-Inspection SR ↑	Search SR ↑	Recovery SR ↑	Dist. (m) ↓	Time (s) ↓	Inspect. ↓
Last Seen + Search	17.2	26.6	12.8	56.2	284	2.19
Time Frequency	21.9	31.3	14.0	52.9	277	2.05
Retrieval + Reasoning	26.6	37.5	17.1	50.3	281	2.03
EVOLVINGNAV	**34.4**	**48.4**	**24.3**	**43.8**	**248**	**1.84**

Across these evaluations, the gains appear at three levels: cross-benchmark results improve initial destination selection, N4 isolates recovery under execution-time motion, and robot trials translate the same belief interface into shorter paths and fewer inspections. Together, these results suggest that the gains extend beyond benchmark-specific exploration or controller choice.

Temporal structure. Under paired controls, the SR gain over Last Seen is 22.36 points in routine worlds but 6.93 points under random transitions (Table 15). The agent trails Last Seen in static worlds and Direct Transformer under random transitions, locating its strongest advantage in learnable temporal regularity. Across five frozen VLMs, paired Search SR gains over Last Seen range from 14.91 to 25.93 points (20.90 on average; Table 12). Matched tools, prompts, episodes, and budgets support transfer across controllers.

5.3 Ablation Studies

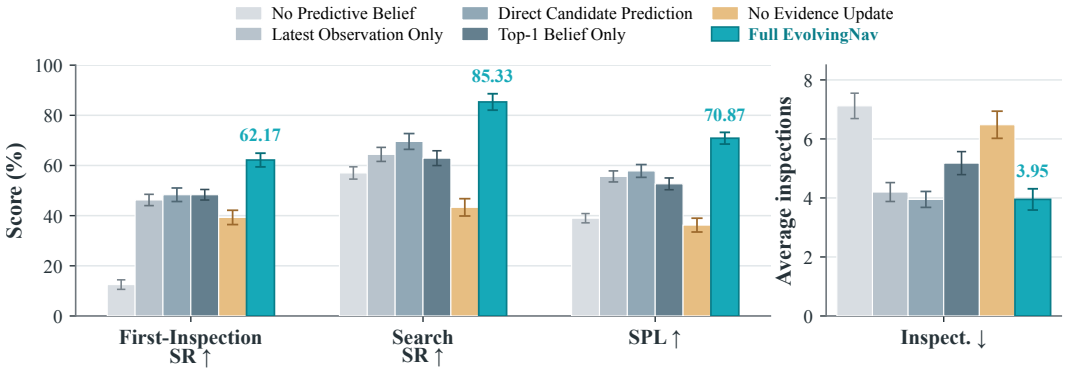

Figure 6: Core component ablation; error bars show five-seed standard deviations.

Core components. The full agent achieves First-Inspection SR of 62.17%, Search SR of 85.33%, and SPL of 70.87%, with 3.95 inspections (Figure 6). Without predictive belief, First-Inspection SR falls to 12.51%; Latest Observation Only reduces Search SR by 20.91 points, indicating that persistent history informs both initial destination selection and recovery. Direct candidate prediction reduces First-Inspection/Search SR by 13.84/15.75 points, supporting the persistence–relocation factorization. Top-1-only search reduces Search SR by 22.41 points, while removing evidence updates reduces it by 42.02 points and increases inspections to 6.48. These results support retaining alternative hypotheses and revising them online to improve search within a fixed action budget. The pattern separates the roles of the components: predictive belief chooses where inspection begins, the full distribution preserves fallback hypotheses, and evidence updating converts new views into route changes. Because No Evidence Update uses the same predictor, its gap isolates the action-time filter rather than improved offline state classification.

Evidence update and calibration. Hard Removal degrades all four metrics (Table 4; Top-1 in percent). Our calibrated update achieves the best rank, Top-1, and ECE, although uncalibrated Bayesian updating has slightly lower NLL. These results support soft, visibility-conditioned evidence over irreversible removal after a missed detection.

Table 4: Negative-evidence update ablation.

Update rule	Rank ↓	Top-1 ↑	NLL ↓	ECE ↓
No Update	5.82	62.98	2.838	0.172
Hard Removal	10.09	42.69	11.514	0.204
Bayesian Update	5.25	64.71	**1.497**	0.151
Bayesian + Calibration	**4.96**	**66.93**	1.530	**0.130**

Prediction quality and transfer. Relative to the Direct Transformer, our predictor improves Top-1 by 4.08/9.47 points and reduces NLL by 0.2367/0.3379 on temporal/LOSO splits. It also exceeds the semantic prior, supporting temporal structure beyond location frequency. The larger LOSO gain supports transfer beyond scene-specific destination statistics, while the lower NLL indicates improved probabilistic prediction. Additional ablations are in Appendix C.

Table 5: Current-state prediction on temporal and LOSO splits.

	Temporal			LOSO		
Predictor	**Top-1 ↑**	**MRR ↑**	**NLL ↓**	**Top-1 ↑**	**MRR ↑**	**NLL ↓**
Last Seen	0.00	0.1699	3.4011	50.00	0.5850	1.8825
Time Frequency	33.85	0.5462	1.8400	49.53	0.6166	1.6455
Direct Transformer	34.16	0.5586	1.7292	51.71	0.6884	1.3157
Shared semantic prior	36.02	0.5957	1.5766	59.47	0.7462	1.0448
Ours	**38.24**	**0.5971**	**1.4925**	**61.18**	**0.7523**	**0.9778**
	±1.42	±0.0291	±0.0867	±1.53	±0.0176	±0.0228

6 Conclusion

We study persistent navigation under unobserved changes before and during execution. EvolvingNav couples a persistence–relocation belief from 4D histories with an event-driven predict–observe–replan filter. Across EvoWorld-Bench, external benchmarks, and robot trials, it improves initial inspection and recovery, especially under learnable temporal structure. Future work will address interacting objects and continuously changing goals. More broadly, the results suggest that persistent embodied memory should be evaluated not only by what it stores, but by whether it supports calibrated inference and evidence-seeking action when the remembered world is no longer current.

REFERENCES

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

Francesco Argenziano, Miguel Saavedra-Ruiz, Sacha Morin, Charlie Gauthier, Daniele Nardi, and Liam Paull. Flowmaps: Modeling long-term multimodal object dynamics with flow matching. *arXiv preprint arXiv:2606.20209*, 2026.

Ermanno Bartoli, Fethiye Irmak Doğan, and Iolanda Leite. Streak: Streaming network for continual learning of object relocations under household context drifts. In *2025 34th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1550–1557. IEEE, 2025.

Berta Bescos, José M Fácil, Javier Civera, and José Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE robotics and automation letters*, 3(4):4076–4083, 2018.

Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *Advances in Neural Information Processing Systems*, 2020.

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

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. *arXiv preprint arXiv:2412.04453*, 2024.

Diane J. Cook, Aaron S. Crandall, Brian L. Thomas, and Narayanan C. Krishnan. CASAS: A smart home in a box. *Computer*, 46(7):62–69, 2013.

Hongyu Ding, Sizhuo Zhang, Ziming Xu, Jinwen Guo, Hongxiu Liu, Xingzhi Cheng, Zixuan Chen, Haifei Qi, Duo Wang, Hao Xu, et al. Uni-lavira: Language-vision-robot actions translation for unified embodied navigation. *arXiv preprint arXiv:2605.27582*, 2026.

Vishnu Sashank Dorbala, Bhrij Patel, Amrit Singh Bedi, and Dinesh Manocha. Personalized embodied navigation for portable object finding. *arXiv preprint arXiv:2403.09905*, 2026.

Muhammad Fadhil Ginting, Sung-Kyun Kim, David D Fan, Matteo Palieri, Mykel J Kochenderfer, and Ali-akbar Agha-Mohammadi. Seek: Semantic reasoning for object goal navigation in real world inspection tasks. *arXiv preprint arXiv:2405.09822*, 2024.

Nicolas Gorlo, Lukas Schmid, and Luca Carlone. Describe anything anywhere at any moment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35002–35013, 2026.

Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5021–5028. IEEE, 2024.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10608–10615, 2023.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37154–37163, 2026.

Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X. Chang, and Manolis Savva. Habitat synthetic scenes dataset (HSSD-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024a.

Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Théophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. GOAT-Bench: A benchmark for multi-modal lifelong navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b.

Jeonghwan Kim, Jisoo Kim, Jeonghyeon Na, and Hanbyul Joo. Parahome: Parameterizing everyday home activities towards 3d generative modeling of human-object interactions. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1816–1828. IEEE, 2025.

Andrey Kurenkov, Michael Lingelbach, Tanmay Agarwal, Emily Jin, Chengshu Li, Ruohan Zhang, Li Fei-Fei, Jiajun Wu, Silvio Savarese, and Roberto Martín-Martín. Modeling dynamic environments with scene graph memory. *arXiv preprint arXiv:2305.17537*, 2023.

Kailing Li, Tianwen Qian, Lijin Yang, Yuqian Fu, Jingyu Gong, Xiaoling Wang, and Liang He. Bridging the 2d-3d gap: A hierarchical semantic-geometric map for vision language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15243–15252, June 2026a.

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

Peiqi Liu, Zhanqiu Guo, Mohit Warke, Soumith Chintala, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Dynamem: Online dynamic spatio-semantic memory for open world mobile manipulation. *arXiv preprint arXiv:2411.04999*, 2024a.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024b.

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

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.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16488–16498. IEEE, 2024.

Manjunath Narayana, Andreas Kolling, Lucio Nardelli, and Phil Fong. Lifelong update of semantic maps in dynamic environments. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6164–6171. IEEE, 2020.

Maithili Patel and Sonia Chernova. Proactive robot assistance via spatio-temporal object modeling. In *Proceedings of the 6th Conference on Robot Learning*, pp. 881–891, 2023.

Maithili Patel, Aswin Prakash, and Sonia Chernova. Predicting routine object usage for proactive robot assistance. In *Conference on Robot Learning*, 2023.

Toby Perrett, Ahmad Darkhalil, Saptarshi Sinha, Omar Emara, Sam Pollard, Kranti Parida, Kaiting Liu, Prajwal Gatti, Siddhant Bansal, Kevin Flanagan, Jacob Chalk, Zhifan Zhu, Rhodri Guerrier, Fahd Abdelazim, Bin Zhu, Davide Moltisanti, Michael Wray, Hazel Doughty, and Dima Damen. HD-EPIC: A highly-detailed egocentric video dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Miguel Saavedra-Ruiz, Charlie Gauthier, Kumaraditya Gupta, Shima Shahfar, Kirsty Ellis, Steven Parkison, and Liam Paull. Predictive spatio-temporal scene graphs for semi-static scenes. *arXiv preprint arXiv:2605.00121*, 2026.

Koya Sakamoto, Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Map-based modular approach for zero-shot embodied question answering. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10013–10019. IEEE, 2024.

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 *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9338–9346. IEEE, 2019.

Tin Stribor Sohn, Maximilian Dillitzer, Jason J. Corso, and Eric Sax. R^4: Retrieval-augmented reasoning for vision-language models in 4d spatio-temporal space. *arXiv preprint arXiv:2512.15940*, 2025.

Hongcheng Wang, Jinyu Zhu, and Hao Dong. User-centric object navigation: A benchmark with integrated user habits for personalized embodied object search. *arXiv preprint arXiv:2602.06459*, 2026.

Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. *Robotics: Science and Systems*, 2024.

Zihao Xin, Wentong Li, Yixuan Jiang, Ziyuan Huang, Bin Wang, Piji Li, Jianke Zhu, Jie Qin, and Shengjun Huang. Agentvln: Towards agentic vision-and-language navigation. *arXiv preprint arXiv:2603.17670*, 2026.

Karmesh Yadav, Yusuf Ali, Gunshi Gupta, Yarin Gal, and Zsolt Kira. Findingdory: A benchmark to evaluate memory in embodied agents. *arXiv preprint arXiv:2506.15635*, 2025.

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.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *arXiv preprint arXiv:2412.06224*, 2024a.

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

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.

Kaiyu Zheng, Rohan Chitnis, Yoonchang Sung, George Konidaris, and Stefanie Tellex. Towards optimal correlational object search. In *IEEE International Conference on Robotics and Automation*, 2022.

Haotian Zhou, Xiaole Wang, He Li, Zhuo Qi, Jinrun Yin, Haiyu Kong, Jianghuan Xu, and Huijing Zhao. Lagmemo: Language 3d gaussian splatting memory for multi-modal open-vocabulary multi-goal visual navigation. *arXiv preprint arXiv:2510.24118*, 2025.

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

Filippo Ziliotto, Tommaso Campari, Luciano Serafini, and Lamberto Ballan. Tango: training-free embodied ai agents for open-world tasks. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24603–24613. IEEE, 2025.

A Additional Method Details

This section presents the episode interface, memory representation, baseline specifications, and implementation details supporting the main method.

A.1 Formal Episode Interface

Each navigation episode references one time-indexed prediction query. Public fields include schema version, episode and query identifiers, split, task type, scene and household identifiers, world variant, target description, query time, agent start pose, candidate-state identifiers, budgets, and success criteria. Evaluation-private fields include the current state, target pose, valid goal viewpoints, hidden transition class, oracle path, mobility label, and query-time activity. Train and validation releases may expose analysis labels in metadata; the formal test release replaces them with null values and computes grouped metrics in a private evaluator.

The high-level track exposes

`NAVIGATE_TO(state)`, `INSPECT(state)`, `EXPLORE`, `STOP`, `NOT_FOUND`,

with a fixed navigation backend. The low-level track exposes Habitat-style forward, turn, look, and stop actions. The main paper uses the high-level track to isolate predictive state selection; low-level results measure sensitivity to control and perception.

The observation history available at query time t_q is

$$\mathcal{H}_{\leq t_q} = \{(I_k, D_k, \xi_k, t_k) \mid t_k \leq t_q\}, \tag{16}$$

where I_k, D_k, ξ_k denote RGB, depth, and camera pose. The task objective in Equation (2) balances successful target verification against travel and inspection costs. N1–N3 fix the target after the query to isolate query-time inference and search, whereas N4 allows additional hidden transitions while the agent moves.

A.2 Detailed 4D Memory Construction

Figure 7 illustrates how observations are converted into persistent entity versions. Each observation contributes spatial evidence and temporal validity, allowing the memory to preserve both previous and current states. Memory is materialized causally from versioned deltas as defined in Equation (3), retaining semantic–spatial relations, time-valid entity versions, and observation provenance. Persistent identity is separated from mutable state: an observed change appends a time-valid version without erasing earlier states, and each memory query uses only evidence available at its decision time.

For image coordinate $\widetilde{\mathbf{u}} = [u, v, 1]^\top$ with depth $D_t(u, v)$, the corresponding point in world coordinates is

$$\mathbf{x}^w = \mathrm{proj}_3\left(\mathbf{T}_{w\leftarrow c,t}\begin{bmatrix} D_t(u,v)\mathbf{K}^{-1}\widetilde{\mathbf{u}} \\ 1 \end{bmatrix}\right), \tag{17}$$

where $\mathbf{K}$ is the camera intrinsic matrix and $\mathbf{T}_{w\leftarrow c,t}$ is the camera-to-world transform. Cross-view association uses semantic compatibility, 3D overlap, temporal consistency, and source confidence. The m-th version of entity o is

$$h_o^m = ([t_{o,s}^m, t_{o,e}^m), s_o^m, c_o^m, f_o^m, \mathcal{P}_o^m), \tag{18}$$

containing a validity interval, candidate state, confidence, visual feature, and evidence handles. An observed state change appends a version without erasing the earlier trajectory, preserving both time-ordered history and observation provenance.

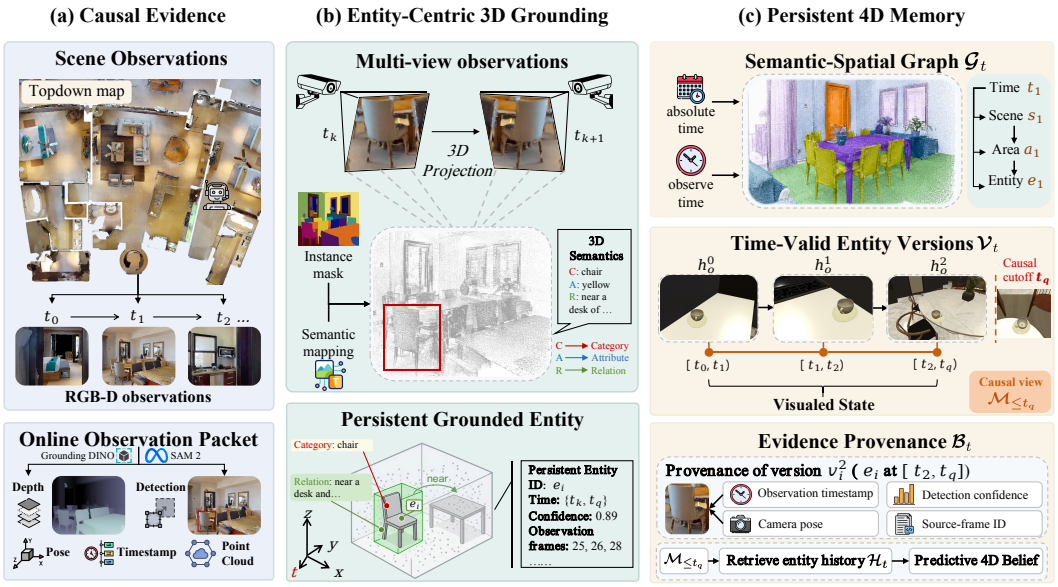

Figure 7: Temporally versioned 4D memory construction.

A.3 Belief Encoding and Training

The retrieved history and event representation are defined in Equations (4) and (5); the Transformer in Equation (6) produces the query representation used by both prediction heads. Event tokens retain entity, state, evidence type, confidence, and visibility alongside elapsed-time and calendar features. The query token specifies the target, query time, and last positive state. The persistence probability in Equation (7) measures agreement with that last positive observation, including departures followed by returns. The shared pointer head in Equation (8) scores every alternative, including unknown, from the query and candidate representations. The candidate encoder combines context, functional role, geometry, and instance identity. The persistence and pointer heads jointly define the normalized belief in Equation (9), trained with current-state negative log-likelihood without a separate change label or oracle mobility type.

A.4 Evidence Updates and Online Prediction

Transition-kernel parameterization and training. The transition operator in Equation (10) is not inferred from a single marginal belief. It is an additional row-conditional prediction head that shares the continuous-time history encoder and candidate embeddings with the query-time predictor. The training set contains chronological tuples $(H_{\le t_a}, s_{t_a}, \Delta t, s_{t_a+\Delta t})$ extracted only from training world histories. For each tuple, the known source state selects one row and the future state supervises

$$\mathcal{L}_{\text{trans}} = -\log[K_\theta(\Delta t \mid H_{\le t_a})]_{s_{t_a}, s_{t_a+\Delta t}}, \qquad \mathcal{L} = \mathcal{L}_{\text{state}} + \lambda_K \mathcal{L}_{\text{trans}}. \tag{19}$$

Prediction horizons are sampled from the empirical range of action-chunk durations and candidate arrival times. The softmax over destination states makes every row sum to one. No transition label, future state, or mobility tag is available at test time. In N1–N3, where the target is fixed after the query, K is the identity; learned propagation is activated only for N4.

Decision epochs and ETA. The filter begins with $b_0^- = b_q^-$ and $b_0^+ = b_0^-$ when no query-time measurement is available. An epoch is triggered after a short action chunk, when a view adds sufficient candidate coverage, when the route or ETA changes materially, when target evidence is obtained, or when the robot reaches an inspection viewpoint. After a chunk of observed duration $\Delta t_j = t_{j+1} - t_j$, Equation (11) first produces the prior at the new current time. For candidate i, the navigation backend then estimates

$$\widehat{\tau}_{j,i} = \frac{d_{\text{geo}}(x_j, \mathcal{W}_i)}{\widehat{v}_j} + \tau_{\text{inspect}}(i), \qquad \widehat{v}_j = \alpha \widehat{v}_{j-1} + (1-\alpha)\frac{\Delta d_j}{\Delta t_j}. \tag{20}$$

All ETAs are recomputed after obstruction or replanning. Candidate-specific arrival beliefs forecast future states from the current posterior. They do not replace the current-time filtering state b_j^+ unless the corresponding prediction horizon has elapsed.

Complete event-driven filter. The implementation follows the same ordering at every decision epoch:

1. Initialize $t_0 = t_q$, $b_0^- = b_q^-$, incorporate any valid query-time measurement once to obtain b_0^+, and initialize an evidence ledger for every candidate.
2. From b_j^+, forecast each $b_{j,i}^{\mathrm{arr}}$ to its own ETA, score known candidates and EXPLORE, and select the highest-utility action.
3. Execute only a short action chunk and record its actual elapsed time, displacement, pose, and RGB-D evidence.
4. Propagate to the new current time: $b_{j+1}^- = b_j^+ K_\theta^{(j)}(t_{j+1} - t_j)$.
5. Create measurements only for candidates whose evidence ledger identifies new information; apply Equation (14) once per evidence identifier to obtain b_{j+1}^+. Stop on a verified positive detection.
6. Update speed and ETAs, forecast all candidate-specific arrival beliefs from b_{j+1}^+, reopen eligible states, and replan if the preferred action changes; otherwise execute the next short chunk.

This explicitly separates the current-time filter $b_j^- \to b_j^+$ from the counterfactual arrival forecasts $\{b_{j,i}^{\mathrm{arr}}\}_i$.

Time-valid evidence rounds without leakage. Candidate i maintains an evidence-round index $m(i)$ and $C_j^{(m)}(i)$, the union of surface samples observed within that round. A no-detection event is created only if the previously unused coverage satisfies $\Delta C_j^{(m)}(i) > \delta_C$, with $\delta_C = 0.05$. Candidate surface samples from 4D memory are projected into the live camera, and online depth marks samples as visible when they lie in the frustum and agree with the measured depth. The resulting online-estimated coverage and other view features form $\mathbf{f}_{j,i}$ for the calibrated detector model g_ϕ in Equation (13). Crucially, $\widehat{r}_{j,i}$ is evaluated on the newly admitted surface samples and rays, not on the entire overlapping image. One view can update several candidates with different strengths. Opportunistic views and deliberate inspections use the same rule, although the latter usually provide more coverage. A candidate is counted as a sufficiently covered inspection when $C_j^{(m)}(i) \geq \tau_{\mathrm{cov}} = 0.70$; this designation does not permanently remove it from the candidate set.
Every measurement has a unique evidence identifier and is incorporated once. Its RGB-D frame and pose remain in memory as provenance, but its likelihood is not multiplied again or reintroduced as an independent history token into the current filter. Ground-truth target masks, poses, unoccluded fractions, and simulator visibility flags are retained only by the private evaluator; an oracle-visibility diagnostic is reported separately from fair comparisons. Incremental masking and evidence identifiers do not assert that video frames are statistically independent; they prevent direct reuse of the same surface evidence. The conditional-measurement model approximates the remaining dependence, with detection probabilities calibrated on validation observations. We never multiply repeated whole-view detection probabilities from overlapping frames.

Reopening candidates and executing unknown. There is no permanent exclusion set. Each candidate stores cumulative coverage, the time of the latest clear negative observation, and cumulative detection probability. After propagation, a previously inspected candidate becomes eligible when its belief exceeds $\tau_b = 0.10$, predicted return probability

$$P_{\mathrm{return}}(i) = \sum_{r \neq i} b_j^+(r) K_\theta^{(j)}(r, i; \Delta t) \tag{21}$$

exceeds $\tau_{\mathrm{return}} = 0.05$, or a new viewpoint offers $\Delta C(i) > \delta_C = 0.05$. Thus a weak or occluded view supports multi-view reinspection, and a location verified to be empty can regain probability mass after sufficient world evolution.

Reopening by a new viewpoint alone retains the current round, so only newly observed geometry contributes evidence. Reopening caused by propagated belief or predicted return after elapsed time starts round $m(i) + 1$ and resets only the coverage gate for that round; the earlier coverage map and negative observations remain in the provenance ledger. Consequently, the same viewpoint can provide a new measurement after the world may have changed, while adjacent frames within one state-validity interval cannot repeatedly suppress the same hypothesis.

The unknown state invokes an executable EXPLORE action with

$$U_j(\mathrm{EXPLORE}) = \frac{b_j(\mathrm{unknown})\widehat{\eta}_j^{\mathrm{discover}}}{c_j^{\mathrm{explore}} + \lambda_{\mathrm{scan}}}. \tag{22}$$

The controller selects a semantic frontier, uncovered region, or unopened container, performs an open-vocabulary scan, adds discovered states to $\mathcal{C}_o$, redistributes unknown mass, and replans. `NOT_FOUND` is allowed only after the exploration budget is exhausted (or no valid frontier remains) and both unknown mass and remaining searchable mass fall below fixed validation thresholds. The same filter supports N1–N4 without a static/dynamic policy router.

A.5 Full Baseline Specification

Last Seen assigns all mass to the latest positive state. Frequency Prior estimates $P(s \mid o)$ from training data. Markov Transition estimates $P(s_{k+1} \mid s_k, o)$; Time-Conditioned Prior additionally conditions on public hour and weekday bins. Instance Hotspot uses only the training trajectory of the target instance. GRU Direct and Transformer Direct receive the same event tokens and candidate encoder as P4D-Nav but directly normalize candidate logits without the persistence factorization. P4D-Belief Prior uses Equation (9) once at the start of each episode. Full EvolvingNav applies Equations (12), (14) and (15) at event-driven decision epochs.

The Oracle Activity diagnostic may use the hidden activity and source location but is excluded from fair rankings. Oracle Current State receives the private target state and measures remaining perception and navigation error. External predictive methods are adapted only through their public inputs; any use of privileged simulator state must be identified and excluded from the main comparison.

A.6 Implementation Details

Prediction model and optimization. The history encoder uses three Transformer layers, hidden width 128, four attention heads, maximum history length $K = 64$, and dropout 0.1. Candidate encoders are shared across scenes. The transition scorer f_θ is a two-layer MLP of width 128 with GELU activation. The history and candidate encoders are shared; the persistence, relocation, and transition heads have separate parameters. We set $\lambda_K = 1$ in Equation (19). We optimize the model with AdamW using a learning rate of 3×10^{-4}, a weight decay of 10^{-2}, and a batch size of 64 for at most 100 epochs. Gradients are clipped to an ℓ_2 norm of 1.0. Early stopping uses a patience of 10 epochs based on validation NLL, and the checkpoint with the lowest validation NLL is retained. We use random seeds $\{0, 1, 2, 3, 4\}$. Experiments use Habitat-Sim 0.3.3 and Habitat-Lab 0.3.3 on $3\times$ NVIDIA GeForce RTX 5080 GPUs.

Perception and controller. The frozen perception stack uses Grounding DINO (Liu et al., 2024b) with a box threshold of 0.35 and a text threshold of 0.25, followed by SAM 2 (Ravi et al., 2025) only for mask refinement. The main high-level agent uses GPT-5.6-Luna as a frozen zero-shot VLM controller, without navigation-task fine-tuning. Candidate viewpoints and low-level planning are fixed in the high-level track; a Habitat low-level action track evaluates the complete embodied stack.

Detection calibration and information boundaries. The function g_ϕ in Equation (13) is a lightweight logistic calibrator that takes as inputs online-estimated candidate coverage, range, viewing angle, projected size, image quality, category, and validation-set detector recall. It is fitted only on validation observations and remains frozen throughout test evaluation. The agent never accesses ground-truth target masks, ground-truth target poses, unoccluded target fractions, or simulator visibility flags.

All architecture choices, optimization hyperparameters, calibration models, perception thresholds, policy thresholds, and checkpoint-selection rules were determined exclusively from the training and validation splits. The test split was not used for model selection or hyperparameter tuning and was accessed only for final evaluation.

A.7 Experimental Details

Benchmark protocols. We evaluate the complete agent on EvoWorld-Bench, FindingDory, and GOAT-Bench under their native task definitions. FindingDory reports high-level goal selection and conditional low-level execution; GOAT-Bench reports official SR and SPL, with Repeat-SR used only as a marked memory-reuse diagnostic. The cross-domain matrix uses R2R-CE on MP3D, HM3D-OVON on HM3D, native PointNav on Habitat-GS, SAGE-Bench on InteriorGS, and EvoWorld-Bench on HSSD. Scores are never pooled across these heterogeneous tasks, and published values are transferred only when split, sensors, actions, and success rules match.

Result provenance and comparison scope. The entries in Tables 1 and 9 come from two explicitly separated sources. A cell marked ^{PR} is transcribed from the cited paper only when the benchmark

version, evaluation split, metric, and success rule match; the marker applies to that cell rather than to an entire method row. Every unmarked baseline entry is reproduced by us. Missing values are left unreported rather than inferred from another task or checkpoint. Accordingly, Table 1 evaluates memory and prediction under the task definition of each benchmark, whereas Table 9 measures cross-environment system performance under native navigation tasks. Neither table is interpreted as a single architecture-controlled ablation; causal attribution to our predictive belief is instead provided by the paired-world control and component ablations.

Test-time information boundary. All reproduced methods receive only information public in the target benchmark. On EVOWORLD-BENCH, observations are truncated at the current decision time. Count and Markov baselines use only the target category, public time, and latest supported state; structured memories such as HOV-SG and DynaMem replay the same causal RGB-D and poses; predictive models receive public state or edge histories and the legal candidate graph. Released navigation agents consume only their native RGB-D/video window, goal specification, and proprioception. The paper-reported FindingDory cell specifically uses the Qwen2.5-VL-3B checkpoint from the cited release and native 96-frame history; it is not recomputed with our planner. No reproduced method receives the private current state, future observations, query-time activity, target pose, oracle viewpoint, or simulator visibility flags. Oracle rows, where present, are diagnostics and are excluded from fair rankings.

Training and model selection. Non-parametric baselines are estimated from the public training split, with smoothing and time-bin settings selected on the validation split. Internal neural baselines use the same event tokens, candidate encoder, training scenes, and validation rule as P4D-NAV. External predictive architectures that require target-domain fitting are trained only on EVOWORLD-BENCH days 0–79 and selected on days 80–84; days 85–89 remain test-only. When an official checkpoint exists for a native benchmark, we retain that checkpoint and its prescribed observation window. Task-adapted systems are labeled separately from zero-shot systems in Table 9; interface conversion is not counted as navigation-policy training.

Prediction splits. Each scene history follows a 90-day chronological split: days 0–79 are used for training, days 80–84 for validation, and days 85–89 for testing. Temporal evaluation pools the held-out test periods and reports 319 changed-only queries; leave-one-scene-out evaluation rotates the held-out scene while preserving the same chronological boundaries and reports 644 change-balanced queries. All predictors share query identities, time-ordered histories, and legal candidate sets. The main table reports five-seed variation; no future observation or private transition label is available at inference time.

Real-world protocol. LYNX M20 is the primary quantitative platform, with X30 and Lite3 used for transfer. The primary M20 comparison uses 64 matched trials, whereas X30 and Lite3 use matched 32-block transfer subsets. All methods receive the same pre-query histories, candidate locations, start conditions, perception interface, and robot-specific low-level stack. Success requires autonomous online confirmation within three inspections during a 360-second episode. Missed detections, navigation failures, timeouts, and human interventions count as failures. Indoor/outdoor balance, platform specifications, and condition-wise breakdowns are provided in Appendix D.2.

Reporting controls. We use two method-preserving adaptation regimes. Memory-only and prediction-only systems retain their native representation or predictor, map their outputs onto the legal candidate states, and use the shared planner, inspection viewpoints, and action budget; this isolates temporal reasoning from low-level control. End-to-end navigation agents retain their released perception, mapping, and policy, with adapters limited to sensor conventions, task-goal formatting, action vocabulary, and `STOP` semantics. We do not add our VLM planner or predictive belief to those agents. The cross-domain scores therefore measure full-system compatibility, while the controlled EVOWORLD-BENCH and real-robot comparisons support module-level claims. Results are stratified by world variant, staleness, transition class, mobility regime, and generalization split. Paired routine/static/random episodes retain scene, query, and marginal placement factors while changing only the hidden transition mechanism.

The EVOWORLD-BENCH entries in Tables 1, 9, 12 and 13 and Figure 6 use the same N1–N5 test episodes, protocol, action budget, and metric definitions. Full-agent results are obtained from separate stochastic runs on these episodes; small numerical differences therefore do not indicate a change in the test set. Controlled gains are computed within each matched comparison. Routine-world and N4-transition diagnostics condition on specified subsets and are identified separately.

B BENCHMARK CONSTRUCTION AND EVALUATION PROTOCOL

This section documents episode validity, leakage controls, mobility assignment, and the metrics used for all reported comparisons.

B.1 CONSTRUCTION AND AUDIT DETAILS

Design principles. EVOWORLD-BENCH follows four principles: temporal continuity, past-only observability, physical executability, and controlled dynamics. Examples are sampled from persistent scene histories rather than independent placements. Public memories contain only evidence available to the robot by the query time. Candidate states must admit collision-free placements and reachable, visibility-checked inspection viewpoints. Paired worlds hold the scene, target, query, and marginal placement factors fixed while varying the hidden transition mechanism. Together, these controls separate predictive reasoning from scene frequency, route geometry, and privileged simulator access.

Source alignment and accounting. Source records constrain the generator rather than serving as episode templates. We distinguish raw source rows or sequences (N_{raw}), records after normalization specific to each source (N_{std}), distinct normalized records referenced by at least one retained generation rule (N_{used}), and total rule references with reuse allowed (N_{ref}). Table 6 reports counts from the audited consumption manifests; reuse therefore does not inflate the number of independent source observations.

Table 6: Audited source contributions.

Source	$\mathbf{N}_{\mathrm{raw}}$	$\mathbf{N}_{\mathrm{std}}$	$\mathbf{N}_{\mathrm{used}}$	$\mathbf{N}_{\mathrm{ref}}$
CASAS Aruba	1,602,820	20,252	428	480
ARAS	5,184,000	5,080	312	360
HD-EPIC	59,454	10,786	742	864
ParaHome	212 seq.	860	186	216
OPPORTUNITY	869,387	2,551	318	372
HOMER+ (sim.)	65 seq.	1,991	356	420

The raw record unit is a sensor row for CASAS, ARAS, and OPPORTUNITY, an annotation row for HD-EPIC, and a sequence for ParaHome and HOMER+. CASAS events are mapped to the room ontology and split into sessions at inactivity gaps exceeding 900 seconds; the resulting occupancy, time-of-day, and weekday statistics parameterize household schedules. ARAS and OPPORTUNITY sensor streams are segmented into activity intervals and normalized to the shared activity, room, and object vocabularies, supplying complementary activity compatibility and temporal-context statistics. HD-EPIC narration verbs and nouns are mapped through canonical action and object aliases to estimate object–action frequencies, without assigning HSSD destinations. ParaHome annotations and object transforms provide displacement and transition-timing evidence. HOMER+ remains explicitly marked as simulated and contributes only long-horizon activity order and terminal-state priors, not independent human observations or test-time model inputs.

Mapping rules and measured parameters. All adapters emit a common schema containing time, activity, actor context, object category, source state, destination state, and provenance. Measured quantities comprise CASAS occupancy timing, ARAS/OPPORTUNITY activity context, HD-EPIC action–object frequencies, ParaHome displacement and transition timing, and HOMER+ simulated order priors. Legal receptacles, reachable viewpoints, and collision-free placements are derived from HSSD geometry. Category-to-receptacle allowlists, affordance constraints, stochastic exception rates, paired static/routine/random interventions, the 90-day horizon, and query sampling are benchmark-authored and labeled as such. Thus no source trajectory is copied verbatim and no source count is expanded into a claimed number of independent human observations. The final manifest spans 54 scenes and 4,860 scene-days and contains approximately 239.19k dynamic state changes and 803.68k task instances after physical-validity, observability, and leakage audits.

Household generation and behavioral checks. For each HSSD scene (Khanna et al., 2024a), resident schedules, room preferences, orderliness, object ownership, and placement habits are fixed at the household level. Activities, resident region trajectories, and object transitions are generated jointly in chronological order over multiple days; stochastic events, delayed returns, and exceptions prevent deterministic evolution. Household profiles, transition rules, and task templates instantiate executable N1–N5, VLN, and EQA episodes. Every relocation is attributed to an activity, resident transition, object lifecycle event, or tidying event. Personal objects alternate between placed and carried states with owner-specific return habits; irregular objects use affordance-valid destinations without temporal or actor-specific predictability. Audits check that personal objects do not simply track their owners and that actor or activity identity does not predict irregular destinations beyond affordance. The audited configuration yields 42.6 region transitions per resident-day, 6.56 relocations per personal object-day, 52.6% within-region relocations, and zero terminal collisions.

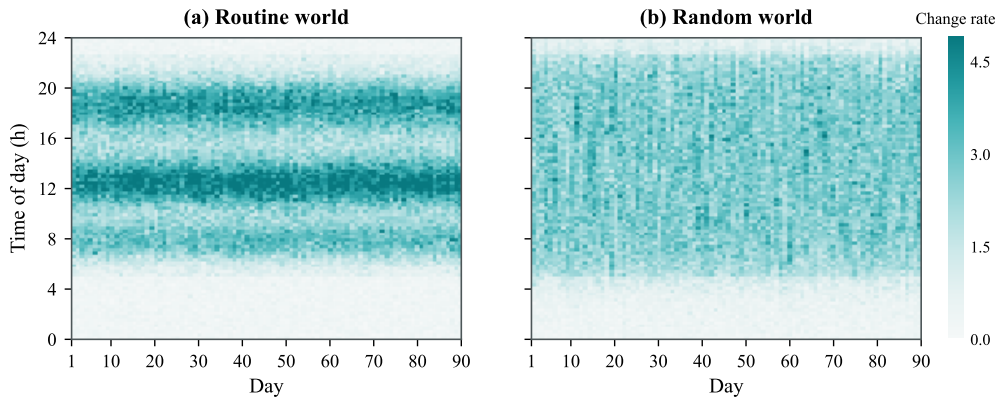

Figure 8: Temporal change density in paired routine and random worlds.

The realized transition density in Figure 8 provides a direct check of the paired-world intervention. Routine worlds retain repeatable, activity-linked time bands, whereas random worlds spread changes across the day while matching the marginal movement process. The control therefore removes predictable temporal structure without changing the 90-day horizon or scene setup. The "Change rate" color scale is the mean number of object relocations per scene-day in each one-hour time-of-day bin.

B.2 Automated Audit and Human Spot Checks

All generated episodes undergo automated quality checks before retention. The checks verify entity identity and event ordering, legal and collision-free placements, reachable candidate viewpoints, consistency between world states and rendered observations, and the query-time cutoff that prevents future information from entering public histories. They also verify target–episode correspondence, valid not-found tracks, and task budgets that permit executable search. Failed episodes are regenerated or excluded; the audit manifest records the check outcomes for every retained episode.

Human assessment uses stratified spot checks rather than reviewing every episode. Five trained researchers sample across data sources, scenes, mobility regimes, and task types. Using a structured checklist, they assess the plausibility of relocation reasons and household routines, the semantic consistency of negative evidence, and the clarity and relevance of language goals. A supervising researcher adjudicates disputed or high-risk sampled cases. Reviewers record accept, revise, or reject decisions for the sampled items; systematic issues prompt corrections to the generation rules and renewed automated checks of the affected episodes. The human review log identifies the sampled items and decisions, while the automated manifest accounts for the full retained set.

Candidate metadata and query sampling. Chronological Habitat replay records navigability, geodesic cost, camera frustum coverage, unoccluded target fraction, and success viewpoints for each candidate. These fields support the physical validity and visibility checks described in Appendix B.4; candidates without collision-free placements and reachable inspection viewpoints are excluded. Query times are sampled from both the natural temporal distribution and windows relevant to the mechanisms under study, including routine stages and intervals after a personal object is dropped. Each navigation query is augmented with a start pose, candidate inspection viewpoints, a path budget, and frozen success conditions. The patrol policy cannot access current object states, resident locations, hidden activities, or generator latents. Query-time activity and hidden transitions are retained only by the evaluator.

Language tasks and release documentation. EQA questions cover existence, location, temporal order, activity association, count, comparison, and compositional relations over the same time-indexed memories. Answers are evaluated against a normalized ontology, and questions requiring future evidence are excluded. VLN replaces the object label with a language goal whose referents and constraints are grounded in the same versioned scene history. The broader task annotations support studies beyond the prediction and navigation experiments emphasized in this paper. Across tracks, only memory representations, current-state beliefs, and resulting decisions differ between controlled methods. Dataset cards and audit summaries document source mappings, authored assumptions, physical validity, candidate coverage, and leakage tests.

Splits and leakage checks. Each 90-day history uses days 0–79 for training, days 80–84 for validation, and days 85–89 for testing. Complete temporal sequences remain within the same split so that no decision uses observations from a later time. Held-out homes are reserved for cross-scene evaluation; paired static/routine/random members remain in the same split. Near-duplicate queries and transition groups are assigned to splits as indivisible units. Serialized releases are checked for future timestamps, private event fields, answer-correlated filenames or indices, repeated event groups across splits, inconsistent paired variants, invalid physics, unreachable goals, and accidental patrol dependence on hidden state. Episode-level checks and mobility assignment are further specified in Appendices B.4 and B.5.

B.3 Mobility Regimes and Task Protocols

Mobility labels are used only for stratified evaluation and are hidden from the agent.

Table 7: Mobility regimes in EvoWorld-Bench.

Regime	Mechanism	Frozen signal	Role
Stable/rare	Infrequent events	Home/return tendency	Persistence control
Routine	Activity state machine	Chain-specific locations	Routine prediction
Personal	Placed/carried lifecycle	Owner placement habit	Long-term memory
Irregular	Affordance sampling	Affordance only	Uncertainty control

Table 8: Prediction and navigation protocols in EvoWorld-Bench.

	Task	Decision signal	Primary measure
P	Current-state prediction	Region/receptacle belief	R@k / NLL
N1	Predictive navigation	Event-filtered belief	First-Inspection SR
N2	Belief-guided search	Full belief + cost	Search SR / cost
N3	Evidence-aware replanning	Updated posterior	Recovery SR
N4	Online-dynamic navigation	Arrival-time belief	Online Recovery SR
N5	Cross-scene generalization	Any protocol above	Held-out SR

B.4 Episode Validity and Leakage Audit

An episode is retained only if: (i) the target has at least one positive pre-query observation; (ii) the target state is known to the private evaluator and physically instantiated without collision; (iii) at least one valid viewpoint lies on the NavMesh; (iv) the start is on the NavMesh, at least 3 meters from the target, and does not reveal it; and (v) all public histories terminate at or before the query. A stop is successful only when the robot is within 1 meter geodesic distance of a valid viewpoint, the private evaluator measures target visible fraction $v^{\mathrm{GT}}_{\mathrm{target}} \geq \tau_{\mathrm{success}} = 0.20$, and the agent identifies the correct target instance or category. This threshold is used only by the private evaluator and is never exposed to the agent.

The three coverage quantities have distinct roles: $\Delta C > 0.05$ triggers a new negative-evidence update, $C \geq 0.70$ defines a sufficiently covered inspection, and $v^{\mathrm{GT}}_{\mathrm{target}} \geq 0.20$ defines evaluation success only. The first two are estimated online from candidate geometry and depth; the third is private evaluator state.

We audit serialized examples for future timestamps, hidden event fields, filenames or indices correlated with answers, duplicate event groups across splits, and inconsistent paired variants. Matched static, routine, and random variants must have identical public query fields. Invalid simulator episodes are reported separately and are not counted as ordinary failures.

B.5 Mobility-Type Assignment

The frozen rule is

$$\mathrm{type}(o) = \begin{cases} \mathrm{stable}, & r_{\mathrm{move}}(o) < \tau_{\mathrm{move}}, \\ \mathrm{activity}, & G_{\mathrm{act}}(o) \geq \tau_{\mathrm{act}}, \\ \mathrm{personal}, & C_{\mathrm{habit}}(o) \geq \tau_{\mathrm{habit}}, \\ \mathrm{irregular}, & \text{otherwise}. \end{cases} \tag{23}$$

All thresholds are selected on training data before navigation evaluation. When the generator specifies a mobility profile, the same statistics are used to verify that realized trajectories exhibit the intended behavior. Every group must cover several categories, instances, and scenes, and no category may uniquely reveal one mobility type.

B.6 METRIC DEFINITIONS

Let S_i be episode success and L_i the executed path. For episodes whose target remains fixed after the query, $L^*_{i,\mathrm{fix}}$ is the shortest feasible path to a valid target viewpoint from the same start. For an online dynamic episode, $L^*_{i,\mathrm{dyn}}$ is the minimum travel distance along a time-feasible trajectory to a viewpoint at which the target can be verified in its scheduled state, under the same start, NavMesh, and action/time budget. The dynamic oracle knows the private transition schedule for evaluation only; the agent does not. Writing L_i^{ref} for the applicable fixed or dynamic oracle path, we use

$$\mathrm{SPL} = \frac{1}{N}\sum_{i=1}^{N} S_i \frac{L_i^{\mathrm{ref}}}{\max(L_i, L_i^{\mathrm{ref}})}. \tag{24}$$

The N1–N5 aggregate includes N4 and applies this reference episode by episode; the fixed-target and online-dynamic slices use their respective reference paths when computing SPL. The static oracle is never applied to a moving target. First-Inspection SR records whether the first candidate to receive sufficient cumulative coverage contains the target; opportunistic evidence and replanning before that inspection are part of the policy. A sufficiently covered en-route view counts as the first inspection itself. Recovery SR conditions on an unsuccessful first inspection and measures whether the target is subsequently found. Excess path uses $L_i - L_i^{\mathrm{ref}}$, with the dynamic oracle for moving-target episodes. Before any N4 rollout, each episode receives a common post-query evaluation window $[t_q, t_q + T_i]$ and a fixed schedule of hidden target transitions. The dynamic subset contains exactly the episode identifiers with a scheduled change of target state inside this window, independent of the destinations, trajectories, or termination times of individual methods. All methods are scored on this same set. The complementary no-transition slice is kept separate. Dynamic SR is the success rate on the preselected dynamic set. Online Recovery SR uses the shared subset whose predeclared transition invalidates the pre-transition target state; only successful verification after that transition counts as recovery, and an early stop does not change the denominator. Revisit Success is the percentage of dynamically reopened candidate visits that recover the target.

We compute calibration metrics on the full candidate belief before navigation and after each update. ECE groups predictions by maximum confidence, the Brier score evaluates all candidates, and true-state rank captures changes beyond Top-1 accuracy. Search-cost curves plot success against the inspection budget and distance traveled, providing comparisons across budgets.

C ADDITIONAL EXPERIMENTAL RESULTS

This section provides extended quantitative results omitted from the main paper because of space constraints.

C.1 CROSS-ENVIRONMENT NAVIGATION

SR and SPL are reported in percent for all environments. Table 9 compares five native benchmark tasks: EVOLVINGNAV leads on seven of ten metrics, including SR and SPL on HM3D-OVON, PointNav, and EVOWORLD-BENCH. Uni-LaViRA leads on R2R-CE SR, while NaVILA leads on R2R-CE and SAGE-Bench SPL. These task-specific exceptions limit any claim of uniform superiority across navigation protocols.

C.2 FULL PREDICTION RESULTS

The temporal split contains 319 changed-only queries and LOSO contains 644 change-balanced queries. We report the mean $\pm$ standard deviation over five seeds for our method; Top-1 is a percentage, while MRR and NLL are unscaled.

Table 9: Cross-environment navigation.

	Standard Simulation Worlds				Photorealistic Simulation				Evolving Simulation Worlds	
	R2R-CE		HM3D-OVON		PointNav		SAGE-Bench		EvoWorld-Bench	
Method	**SR**	**SPL**	**SR**	**SPL**	**SR**	**SPL**	**SR**	**SPL**	**SR**	**SPL**
Navigation-trained or task-adapted										
MTU3D (Zhu et al., 2025)	8.8	6.5	40.8[PR]	12.1[PR]	75.8	72.7	25.9	15.6	35.2	30.7
NaVid (Zhang et al., 2024b)	37.4[PR]	35.9[PR]	60.3	36.0	15.3	13.4	17.4	15.1	31.0	22.6
Uni-NaVid (Zhang et al., 2024a)	47.0[PR]	42.7[PR]	39.5[PR]	19.8[PR]	15.9	13.2	26.4	23.1	55.3	41.7
NaVILA (Cheng et al., 2024)	54.0[PR]	49.0[PR]	18.3	9.7	78.4	77.5	39.0	**34.0**	35.2	30.7
Zero-shot navigation or task adapters										
HSGM (Li et al., 2026a)	47.9[PR]	32.8[PR]	40.0	32.5	37.2	35.3	11.8	9.4	17.0	12.5
VLFM (Yokoyama et al., 2024)	12.0	9.6	35.2[PR]	19.6[PR]	53.0	42.2	24.4	18.8	35.0	25.8
AO-Planner (Chen et al., 2025)	25.5[PR]	16.6[PR]	30.9	22.0	76.6	74.7	27.4	19.7	37.0	29.3
TANGO (Ziliotto et al., 2025)	14.3	9.87	35.5[PR]	19.5[PR]	35.8	33.5	22.7	14.7	31.0	23.3
Uni-LaViRA (Ding et al., 2026)	**60.7[PR]**	**47.7[PR]**	60.0[PR]	40.5[PR]	81.3	77.7	35.6	28.7	65.2	39.4
MSGNav (Huang et al., 2026)	11.6	9.4	48.3[PR]	27.0[PR]	37.2	34.8	23.5	17.5	26.7	19.6
EvolvingNav (ours)	55.8	43.1	**64.2**	**51.7**	**86.2**	**79.8**	**42.4**	32.1	**86.6**	**68.2**

[PR] Paper-reported under the cited native protocol; unmarked baseline scores are our reproductions under the method-preserving protocol in Appendix A.7.

Table 10: Full current-state prediction results.

Predictor	**Temporal Top-1 ↑**	**MRR ↑**	**NLL ↓**	**LOSO Top-1 ↑**	**MRR ↑**	**NLL ↓**
Uniform Random	9.32	0.2607	2.4509	11.96	0.3050	2.2980
Last Seen	0.00	0.1699	3.4011	50.00	0.5850	1.8825
Object Frequency	30.75	0.5476	1.7397	50.47	0.6441	1.8455
Time Frequency	33.85	0.5462	1.8400	49.53	0.6166	1.6455
Markov	18.94	0.4435	2.3772	49.38	0.6286	1.6982
Activity-Markov	25.16	0.4582	2.2436	49.84	0.6221	1.6183
Random Forest	20.50	0.4851	1.9181	58.07	0.7283	1.1441
Packed MLP	29.50	0.5225	2.1036	58.07	0.7173	1.3105
GRU	32.61	0.5471	1.8279	54.04	0.7016	1.2558
Direct Transformer	34.16	0.5586	1.7292	51.71	0.6884	1.3157
Shared semantic prior	36.02	**0.5957**	1.5766	59.47	0.7462	1.0448
Ours	**38.24** ±1.42	**0.5971** ±0.0291	**1.4925** ±0.0867	**61.18** ±1.53	**0.7523** ±0.0176	**0.9778** ±0.0228

C.3 Real-World Search Case

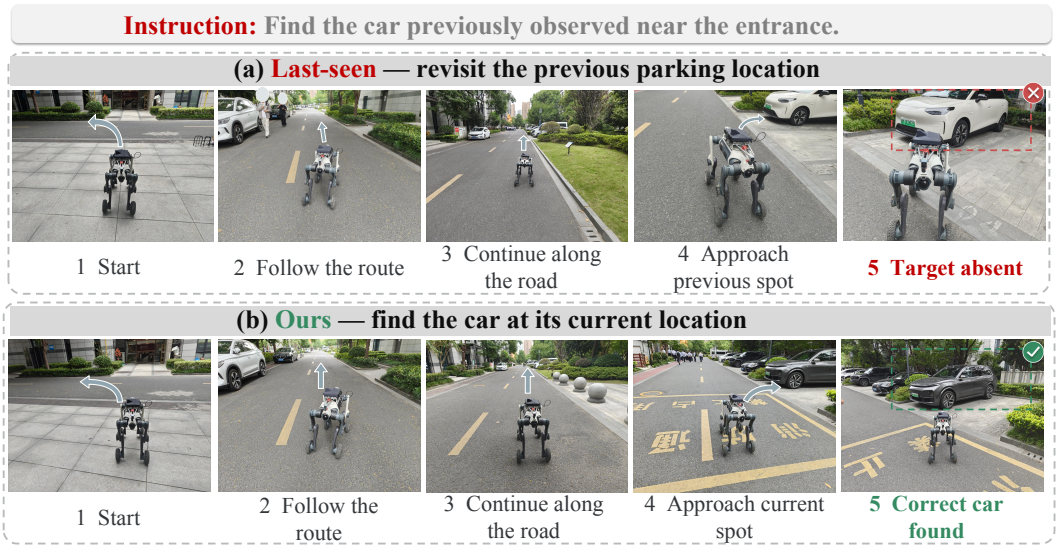

Figure 9: Outdoor car search under stale and predictive memories.

C.4 EvoWorld-Bench Detailed Results

Table 11 complements the main benchmark table with search cost and recovery after an unsuccessful first inspection; lower is better for the two cost columns. Table 12 tests whether paired navigation gains persist across different VLMs under matched tools, prompts, episodes, and action budgets. In the VLM study, SR is reported in percent and SPL is unscaled.

Table 11: EvoWorld-Bench routine-world search cost and recovery.

Method	Inspect. ↓	Distance (m) ↓	Recovery SR (%) ↑
Navigation and persistent memory			
VLFM (Yokoyama et al., 2024)	4.85	18.72	22.41
LagMemo GLUE (Zhou et al., 2025)	3.92	13.24	45.62
HOV-SG (Werby et al., 2024)	3.36	12.08	51.24
DynaMem (Liu et al., 2024a)	2.51	11.59	63.93
Predictive world and state modeling			
SLaTe-PRO (Patel et al., 2023)	4.40	14.58	29.00
SGM+NEP (Kurenkov et al., 2023)	4.08	**7.42**	27.56
FlowMaps (Argenziano et al., 2026)	4.66	16.51	20.27
PredictiveGraphs (Saavedra-Ruiz et al., 2026)	3.71	10.48	17.96
EvolvingNav (ours)	**2.18**	8.36	**68.42**

EvolvingNav requires the fewest inspections (2.18) and achieves the highest Recovery SR (68.42%), improving over DynaMem by 4.49 points after an unsuccessful first inspection. SGM+NEP travels the shortest average distance (7.42 m), while EvolvingNav remains close at 8.36 m. Together with the First-Inspection, Search SR, and SPL results in Table 1, these complementary metrics indicate that retaining uncertainty and updating evidence reduce redundant inspections and support recovery from an incorrect initial prediction.

Table 12: VLM robustness on EvoWorld-Bench.

Different VLMs	Last-seen Agent			EvolvingNav			Δ Search SR ↑
	First-Inspection SR ↑	Search SR ↑	SPL ↑	First-Inspection SR ↑	Search SR ↑	SPL ↑	
GPT-4o	31.88	48.94	0.3215	**59.72**	**71.86**	**0.6027**	**+22.92**
GPT-5.5	42.36	51.31	0.3732	**63.48**	**73.57**	**0.6531**	**+22.26**
GPT-5.6-Luna	46.92	57.43	0.3970	**63.87**	**83.36**	**0.6639**	**+25.93**
Qwen2.5-VL-3B	16.26	26.55	0.1502	**38.27**	**45.02**	**0.3727**	**+18.47**
Qwen2.5-VL-32B	18.84	31.30	0.1591	**42.44**	**46.21**	**0.3713**	**+14.91**
					Average gain		**+20.90**

C.5 Fine-Grained Ablation Analysis

The main-paper ablation isolates the five decisions that define the complete agent. Here we examine implementation choices within temporal memory, evidence updating, and the predictor while preserving the same perception, candidate set, inspection viewpoints, and low-level navigator.

C.5.1 Temporal and Memory Ablations

SR and SPL are reported in percent throughout this analysis.

Table 13: Temporal and memory ablations.

Group	Variant	First-Inspection SR ↑	Search SR ↑	SPL ↑	Inspect. ↓
Temporal encoding	w/o elapsed-time encoding	48.37	80.28	64.82	4.36
	w/o calendar context	47.75	76.11	66.20	4.05
	w/o both temporal cues	40.84	76.03	64.98	4.12
Historical evidence	w/o instance-specific history	57.57	80.77	66.61	4.30
	w/o negative history	45.86	79.53	63.83	4.44
History capacity	$K = 8$	39.55	63.77	56.00	4.57
	$K = 16$	45.88	69.48	65.77	4.37
	$K = 32$	56.25	79.60	68.40	4.15
	$K = 64$ **(Full)**	**60.23**	**84.63**	**70.34**	**3.98**

Elapsed-time and calendar ablations distinguish irregular observation gaps from daily and weekly periodic context. Removing both cues reduces First-Inspection SR by 19.39 points relative to the full

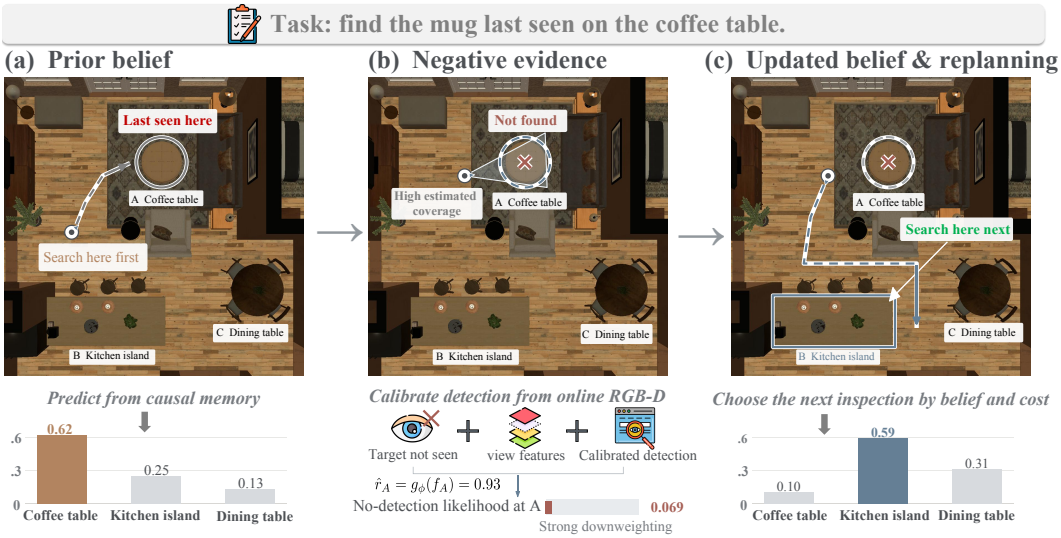

Figure 10: Evidence-aware belief updating and replanning after a negative inspection.

model. Instance-specific trajectories provide smaller but consistent gains, whereas removing negative history reduces First-Inspection SR by 14.37 points and SPL by 6.51 points. Increasing capacity from $K = 8$ to $K = 64$ raises First-Inspection SR by 20.68 points, Search SR by 20.86 points, and SPL by 14.34 points while reducing the average inspection count from 4.57 to 3.98. These trends show that performance benefits from temporally structured evidence over a sufficiently long history, rather than merely the most recent observations.

C.5.2 Evidence Update Ablations

Figure 10 illustrates one deliberate-inspection event in the same stepwise filter used for opportunistic views. Causal memory initially assigns probability 0.62 to the last-seen coffee table. The frozen logistic calibrator maps the online view features $\mathbf{f}_{j,i}$ to $\widehat{r}_{j,i} = 0.931$; when the target is not detected, the likelihood at that state is therefore 0.069. The posterior consequently shifts from $(0.62, 0.25, 0.13)$ to $(0.10, 0.59, 0.31)$ and the planner selects the kitchen island next. The inspected state retains nonzero mass, reflecting residual perceptual uncertainty rather than an irreversible deletion.

Table 4 in the main paper compares the four update rules. No Update retains the prior; Hard Removal assigns zero probability after each unsuccessful inspection; Bayesian Update applies a soft likelihood; and our calibrated variant conditions that likelihood on online view features.

C.5.3 Predictor Architecture

Table 14: Predictor architecture on changed-only queries.

Predictor	Top-1 ↑	MRR ↑	NLL ↓
GRU	32.61	0.5471	1.8279
Direct Transformer	34.16	0.5586	1.7292
Structured Transformer (ours)	$\mathbf{38.24 \pm 1.42}$	$\mathbf{0.5971 \pm 0.0291}$	$\mathbf{1.4925 \pm 0.0867}$

All three predictors use the same observation tokens and candidate encoder. The Direct Transformer normalizes candidate logits directly, whereas the structured model separates persistence from relocation. This controlled comparison therefore isolates architectural factorization from memory content and action selection.

C.6 Evolution-Aware Diagnostics

Paired-world control. The paired-world comparison in Table 15 holds scenes and queries fixed while changing only the world transition mechanism. Last Seen reaches 84.38 SR in static worlds but falls to 38.49 under routine evolution, exposing the failure of stale memory. In contrast, EvolvingNav reaches 60.85 routine-world SR, improving by 22.36 points over Last Seen and by 12.03 points over Transformer Direct. Under matched random motion, the gain over Last Seen decreases to 6.93 points and EvolvingNav trails Transformer Direct by 1.27 points. This contrast localizes its main advantage to learnable temporal regularity rather than a scene-frequency or generic search shortcut.

Table 15: Paired-world SR and routine gain (%).

Method	Static SR	Routine SR	Random SR	Routine gain over Last Seen
Last Seen	**84.38**	38.49	23.19	–
Markov Transition	80.22	32.57	20.64	−5.92
Direct Transformer	69.79	48.82	**31.39**	+10.33
EVOLVINGNAV	82.19	**60.85**	30.12	+22.36

Effect of memory staleness and inspection budget. Figure 11 complements these aggregate tables by showing how performance changes with temporal staleness, inspection budget, and hidden mobility type. These episode-level diagnostics are not inferred from aggregate success rates.

Behavior-shift and trace-grounding diagnostic. Figure 12 tests whether the learned regularities are limited to the nominal timing of the benchmark generator. Online Recovery SR falls for all methods under phase offsets, return delays, destination shifts, and their combination, but EVOLVINGNAV achieves higher recovery than Last Seen and PredictiveGraphs in every condition. The duration CDF further shows that the generated transition durations track the broad temporal scale of held-out real traces without exactly reproducing them. This diagnostic supports robustness to moderate behavior shift and temporal grounding beyond a single generator setting; it is not presented as evidence of unrestricted out-of-distribution generalization.

D REAL-WORLD EVALUATION

This section describes the robot platforms, presents M20 results by environment and temporal condition, and provides qualitative cases.

D.1 ROBOT PLATFORMS

We use three commercial DEEP Robotics platforms: LYNX M20, X30, and Lite3. Platform-specific perception and low-level control use a common interface, while the memory, prediction, belief update, and high-level search components remain unchanged.

Table 16: Robot platforms for real-world evaluation.

Platform	Morphology	Standing size	Mass	Endurance / range
LYNX M20	Wheel-legged	820 × 430 × 570 mm	∼35 kg	3 h / 15 km unloaded
X30	Quadruped	1000 × 695 × 470 mm	56 kg	2.5–4 h / ≥10 km
Lite3 (LiDAR)	Quadruped	610 × 370 × 496 mm	13.5 kg	1.5–2 h / 2.7 km

Manufacturer-rated endurance and range are descriptive specifications, not experimental outcomes. Absolute completion time is compared only between methods executed on the same platform.

D.2 ENVIRONMENT BREAKDOWN

Table 17 separates indoor and outdoor trials. The outdoor setting has longer routes and lower success for all methods, while EVOLVINGNAV improves performance in both environments.

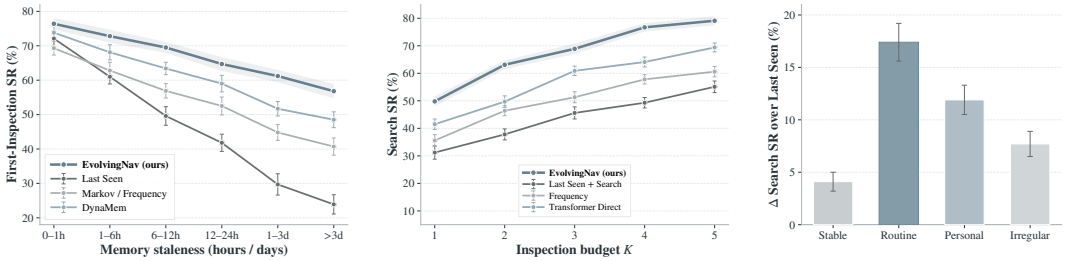

Figure 11: Predictive diagnostics under memory staleness and search budgets.

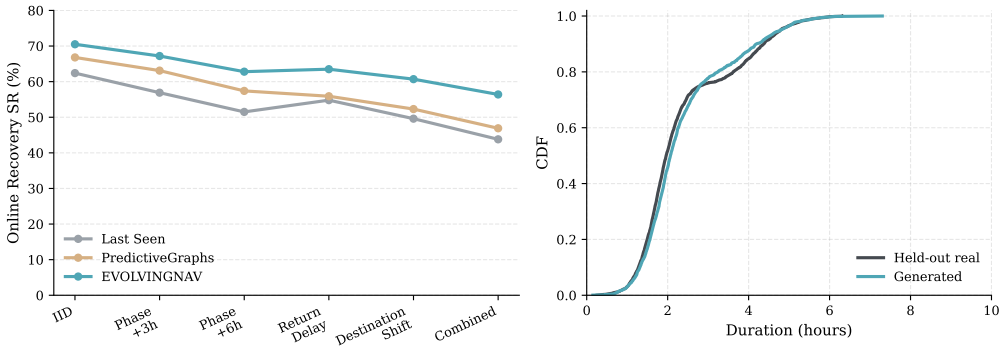

Figure 12: Recovery under behavior shift and temporal grounding to held-out traces.

Table 17: LYNX M20 results by environment.

	Indoor			Outdoor		
Method	**First-Inspection**	**Search SR**	**Dist.**	**First-Inspection**	**Search SR**	**Dist.**
Last Seen + Search	18.8	31.3	28.6	15.6	21.9	83.8
Time Frequency	25.0	34.4	27.2	18.8	28.1	78.6
Retrieval + Reasoning	31.3	40.6	25.1	21.9	34.4	75.5
EVOLVINGNAV	**40.6**	**53.1**	**22.4**	**28.1**	**43.8**	**65.2**

D.3 TEMPORAL CONDITION ANALYSIS

Temporal conditions. Table 18 shows that predictive memory is most useful when recurring changes provide a usable historical signal. Last Seen remains strong when nothing changes, whereas all methods degrade as regularity weakens. This breakdown distinguishes the benefits of routine modeling from improvements attributable to generic search.

Table 18: LYNX M20 Search SR by temporal condition (%).

Method	**Unchanged**	**Routine-consistent**	**Weak-routine**	**Broken-routine**
Last Seen + Search	**68.8**	18.8	12.5	6.3
Time Frequency	50.0	50.0	18.8	6.3
Retrieval + Reasoning	62.5	43.8	25.0	18.8
EVOLVINGNAV	62.5	**68.8**	**37.5**	**25.0**

Recovery and cross-platform transfer. Following unsuccessful first inspections, EVOLVINGNAV recovered 9/37 M20 episodes (24.3%). Table 19 reports the matched 32-block transfer comparison for X30 and Lite3 alongside the corresponding M20 subset; the primary M20 result still uses all 64 trials. This evaluates transfer, not platform parity.

Table 19: Cross-platform navigation transfer.

Platform	**Method**	**First-Inspection SR**	**Search SR**	**Dist. (m)**
LYNX M20	Last Seen + Search	18.8	25.0	55.4
	EVOLVINGNAV	**37.5**	**50.0**	**44.6**
X30	Last Seen + Search	18.8	25.0	54.1
	EVOLVINGNAV	**31.3**	**43.8**	**46.8**
Lite3	Last Seen + Search	18.8	31.3	51.8
	EVOLVINGNAV	**25.0**	**43.8**	**45.9**

D.4 QUALITATIVE CASES

The outdoor comparison in Figure 9 isolates stale memory in a car-search task: Last Seen revisits the previous entrance-side parking location and observes that the target is absent, whereas EVOLVINGNAV reaches its current location. The two-row indoor comparison in Figure 13 shows the same failure mode for a cup: EVOLVINGNAV routes to the predicted office location, while Last Seen first inspects the now-empty pantry rack. These traces visualize representative outcomes and do not add observations to the aggregate success-rate measurements.

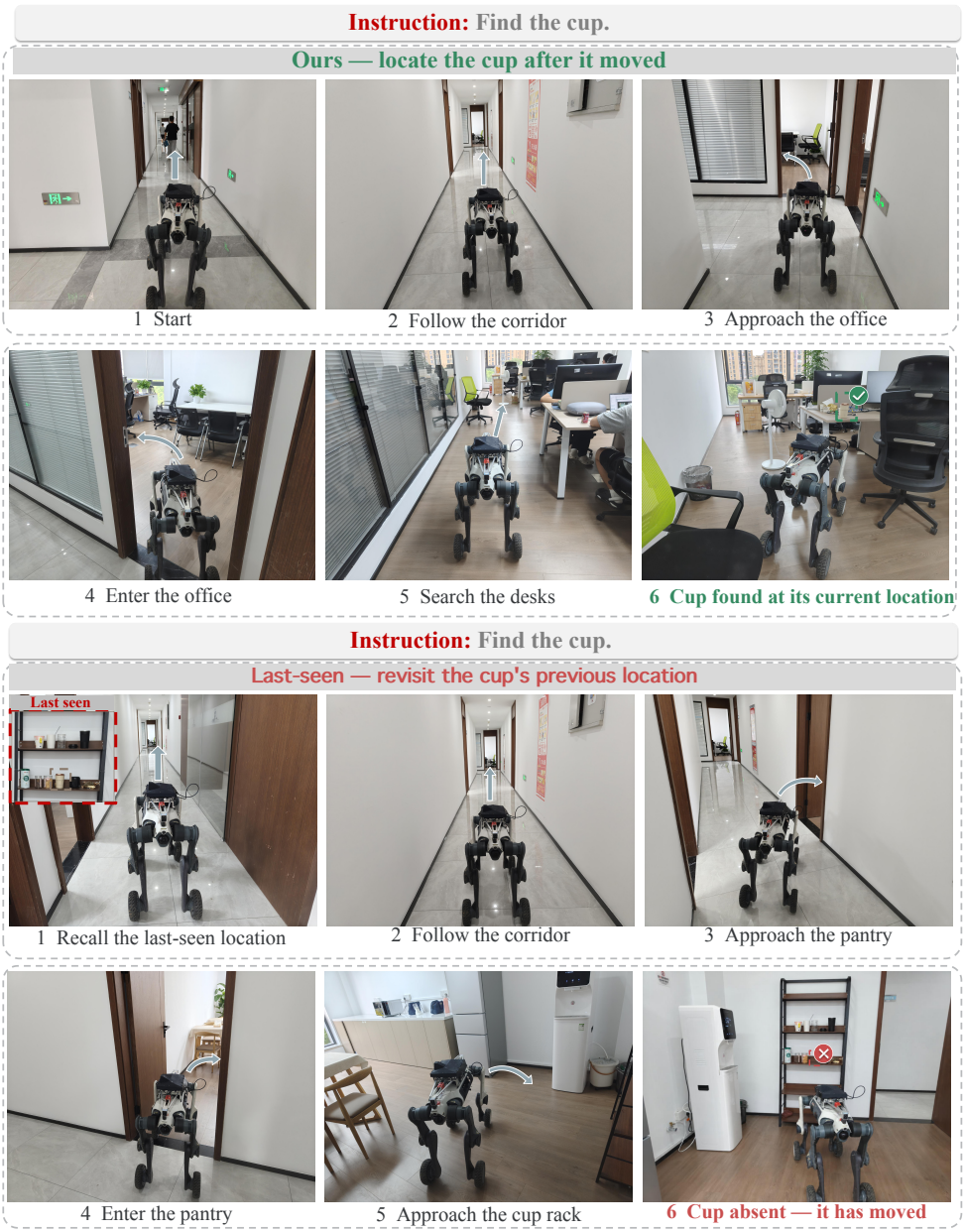

Figure 13: Indoor cup search with predictive belief (top) and stale Last Seen memory (bottom).

Additional system execution cases. Beyond predictive target search, Figures 14 and 15 document two longer-horizon executions. The indoor case combines navigation, person verification, document collection, and delivery. The LYNX M20 has no manipulator, so a human places the documents on the robot after person verification; the robot then autonomously transports them to the meeting room. The outdoor case shows obstacle detection and route replanning during shared-bike search. These cases demonstrate broader system orchestration but are not used as quantitative evidence for the predictive-belief module.



Figure 14: Long-horizon indoor document-delivery execution.

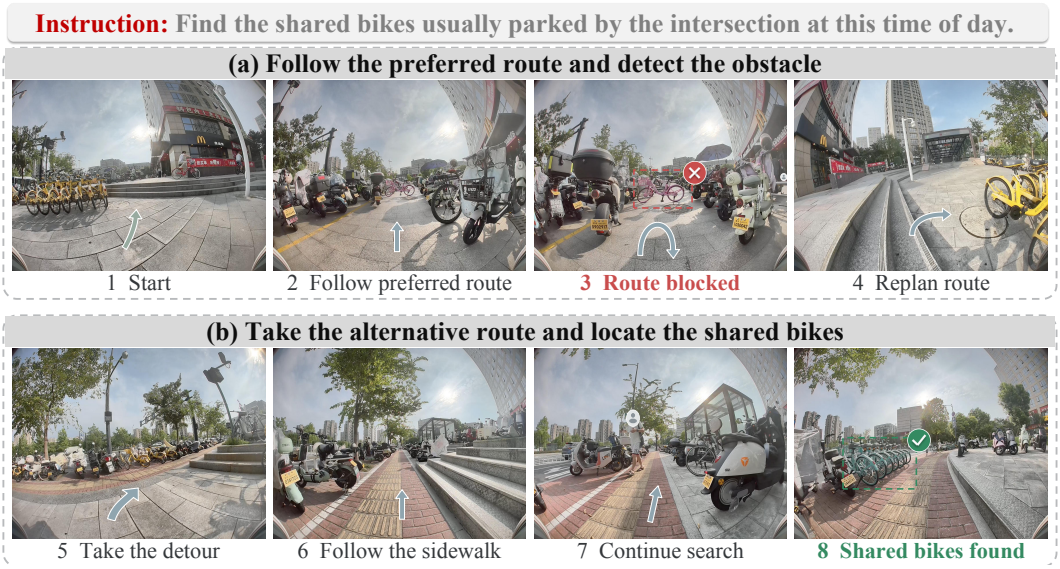

Figure 15: Long-horizon outdoor shared-bike search.

E Reproducibility and Responsible AI

Ethics and AI use. Persistent visual memory can capture people or private spaces, and incorrect beliefs may induce unsafe motion. Deployments should obtain consent, control data retention and access, preserve provenance, and retain platform-specific collision avoidance and emergency stops. The benchmark compiles existing records into simulated HSSD episodes without a new human-subject study. Generative AI assisted manuscript editing, LaTeX checks, and local engineering; it did

not produce benchmark observations or ground-truth labels. The authors reviewed this material and remain responsible for the paper and artifacts.