

WorldLines: Benchmarking and Modeling Long-Horizon Stateful Embodied Agents

Yehang Zhang^{1,3*}, Jianchong Su^{1,3*}, Haojian Huang^{1,3*},
Yifan Chang³, Tianhao Zhou^{1,3}, Xinli Xu¹, Yingjie Xu^{1,3},
Yinchuan Li³, Zexi Li^{4,3‡}, Ying-Cong Chen^{1,2†}

¹ The Hong Kong University of Science and Technology (Guangzhou)

² The Hong Kong University of Science and Technology

³ Knowin AI

⁴ The Chinese University of Hong Kong

Abstract

To assist humans over extended periods in real homes, embodied agents must remember user routines, world states, and past interactions. Existing long-term memory benchmarks mainly evaluate language-centric retrieval and question answering, while embodied benchmarks often focus on short-horizon task execution without testing long-term memory use in dynamic environments. We introduce WorldLines, a project-driven benchmark for long-horizon embodied household assistance. It constructs temporally extended household traces with dialogues, actions, execution feedback, object and device state changes, and converts them into evidence-linked samples for Memory QA and Embodied Task Planning. We further propose ObsMem, an observer-grounded memory framework that maintains visibility-aware memories and action-native state trails for state-aware decisions. Experiments reveal persistent challenges in partial observability, overwritten world states, and translating long-term memory into embodied plans, while ObsMem offers a stronger reference architecture for this setting.

1 Introduction

To operate reliably over long horizons, embodied agents need more than memory of past interactions; they must maintain a stateful view of an evolving world (Fung et al., 2025; Yang et al., 2025; Chu et al., 2026). Real service requests often unfold over time and depend on user routines, object states, device settings, and recent events. For example, a user may say: “I am going to the gym at 7:30 and will be back home at 8:30. After I return, I would like to watch a movie in the living room as usual and have something to eat. I also just bought some fruit and put it in the refrigerator.” Responding correctly requires the robot to connect the current

*Equal contribution. † Corresponding author. ‡ Project leader.

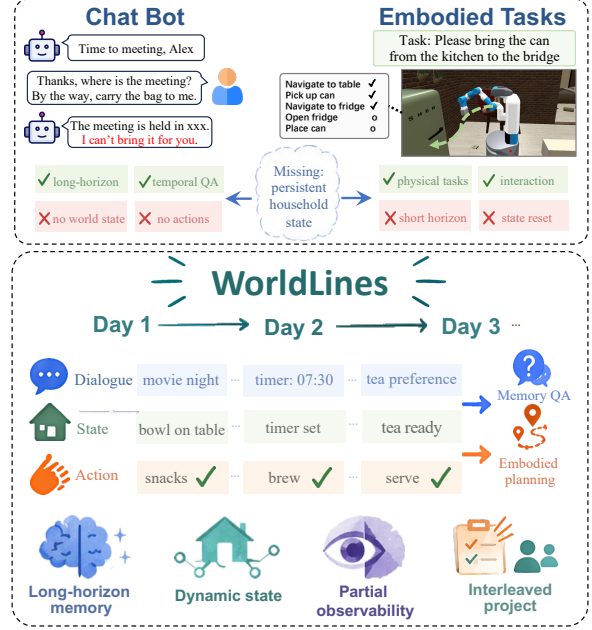


Figure 1: **Overview of WorldLines.** WorldLines tracks cross-day dialogue, state changes, and actions for memory QA and state-aware embodied planning.

instruction with prior schedules, preferences, and environmental states.

This challenge becomes sharper in embodied settings. Long-horizon embodied tasks should not be reduced to isolated dialogues, single action episodes, or one-off state changes. Existing embodied benchmarks have advanced navigation, rearrangement, manipulation, and multi-agent planning (Li et al., 2023; Puig et al., 2024), but they typically remain bounded within short episodes where state does not persist across interactions. Real long-horizon interaction instead requires agents to maintain an evolving world state across dialogue, human activity, robot actions, and device changes. Because the world is partially observable, objects may be moved outside the robot’s view, and container or device states may change without direct observation. Therefore, this work focuses not on

single-task execution, but on whether agents can maintain partially observable world states and use them for later question answering, planning, and execution.

As summarized in Table 1, existing benchmarks split this problem into two incomplete settings. Long-term memory benchmarks evaluate cross-session retrieval, updating, and question answering, but usually decouple memory from physical state transitions, action feedback, and executable constraints (Wu et al., 2025; Maharana et al., 2024). Embodied benchmarks cover navigation, rearrangement, manipulation, and multi-agent planning, but are mostly confined to short episodes, where world states rarely persist across interactions or affect later tasks (Chang et al., 2025; Shridhar et al., 2021). This raises a central question for embodied-agent evaluation: *can agents maintain persistent state over long-horizon, partially observable interactions and use it for downstream embodied tasks?*

This motivates **WorldLines**, a benchmark for evaluating long-horizon stateful embodied agents (Figure 1). WorldLines generates extended traces that include dialogue, human activity, robot actions, device control, execution feedback, and world-state changes, and converts them into evidence-linked Memory QA and Embodied Task Planning samples. In this setting, memory is not the endpoint of evaluation; it is the mechanism through which agents maintain state, trace evidence, and make later decisions.

WorldLines shows that long-horizon embodied agents require more than flat text-retrieval memory. Text-snippet memories (Xu et al., 2026b; Kang et al., 2025; Chhikara et al., 2025; Xu et al., 2026a) struggle to distinguish direct observations, reported information, and unobserved changes, and to track action-induced updates to objects, containers, and devices. We therefore introduce **ObsMem**, an observer-grounded memory framework that separates historical evidence, structured world states, and agent beliefs to support persistent state maintenance and embodied decision making under partial observability.

The main contributions of this work are as follows:

- We introduce **WorldLines**, a benchmark for long-horizon stateful embodied agents, covering Memory QA and Embodied Task Planning in dynamic, partially observable environments.

- We develop a project-driven trace generation pipeline that turns grounded worlds, long-term activity threads, executable actions, and evolving states into evidence-linked evaluation samples.
- We propose **ObsMem**, an observer-grounded memory framework that separates event evidence, state trails, and belief records for long-horizon embodied QA and planning.

2 Related Work

2.1 Benchmarks for Long-Horizon Agent Memory

Long-term memory benchmarks for LLM-based agents have been developed primarily in conversational and multimodal settings. LoCoMo (Maharana et al., 2024) evaluates cross-session dialogue memory, LongMemEval (Wu et al., 2025) studies long-span memory updating, HaluMem (Chen et al., 2026) focuses on memory consistency and hallucination, and RealMem (Bian et al., 2026) introduces project-oriented long-term interaction. These benchmarks provide useful protocols for retrieval, QA, and consistency evaluation, but remain largely text-centric. Embodied benchmarks cover complementary settings: ALF-World (Shridhar et al., 2021) studies language-conditioned household tasks, ProcTHOR (Deitke et al., 2022) and Habitat 3.0 (Puig et al., 2024) support simulated navigation and interaction, while PARTNR (Chang et al., 2025) and BEHAVIOR-1K (Li et al., 2023) focus on collaboration, rearrangement, and long-horizon task execution. EvoEmpirBench (Zhao et al., 2025) further evaluates dynamic spatial reasoning under partial observability, but centers on game-like navigation and elimination tasks. However, they do not explicitly evaluate long-term memory in embodied task completion.

2.2 Agent Memory Systems

Existing LLM-based agents commonly use external memory to store, update, and retrieve information beyond a single context window (Hu et al., 2026b). MemGPT (Packer et al., 2024) organizes context and external storage into hierarchical memory tiers, while MemoryBank (Zhong et al., 2024) accumulates user-specific memories from long-term conversations. Recent systems further improve memory management through operating-system-inspired scheduling (Kang et al., 2025), scalable extraction and update pipelines (Chhikara et al.,

Benchmark	Setting	Long-Term	Project-Driven	Persistent World State	Physical & Device Ops.
LongMemEval	Dialogue	✓	—	—	—
LoCoMo	Dialogue	✓	—	—	—
RealMem	Dialogue	✓	✓	Project state	—
MEMENTO	Embodied	—	—	—	✓
PARTNR	Embodied	—	—	—	✓
WorldLines	Household Sim.	✓	✓	World state	✓

Table 1: Comparison of representative long-term memory and embodied-agent benchmarks. WorldLines combines project-driven long-term memory with persistent household world states, physical actions, and smart-device operations in simulated household environments.

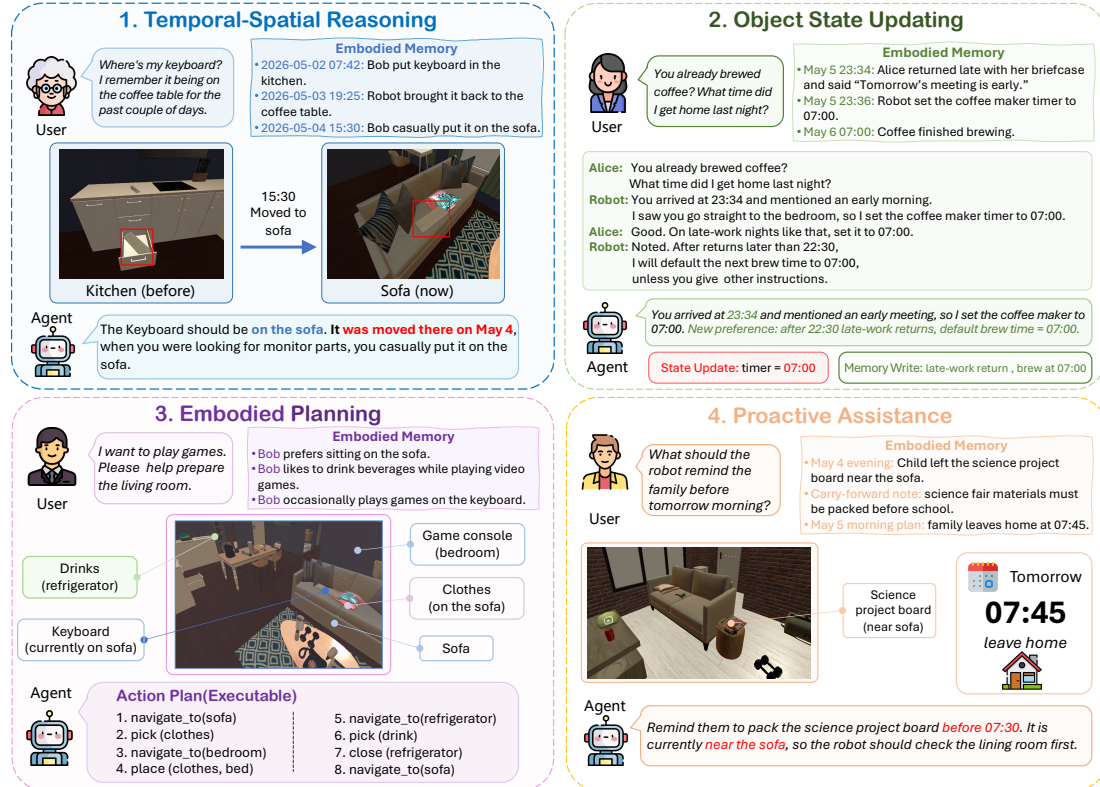


Figure 2: **Core dimensions of WorldLines.** WorldLines evaluates four aspects of long-horizon embodied tasks: temporal-spatial reasoning, object state, embodied planning, and proactive assistance.

2025), agentic memory organization (Xu et al., 2026b), and graph-structured relational memory (Hu et al., 2026a). These works mainly study persistent memory for conversational or general-purpose agents. Memory has also been explored in embodied agents. MEMENTO (Kwon et al., 2025) studies personalized embodied assistance, while semantic-map and scene-graph methods maintain structured object, spatial, and relational knowledge for planning (Rana et al., 2023; Gu et al., 2024). Other approaches retrieve past observations for embodied decision making (Xie et al., 2025; Wang et al., 2025; Zhou et al., 2024; Lillemark et al., 2026) or store reusable skills and programs for future tasks (Wang et al., 2024).

3 WorldLines Benchmark Construction

3.1 Benchmark Formulation

WorldLines evaluates whether embodied agents can maintain household world states over long-term interactions. Each sample is derived from a multi-day household trace containing dialogue, human activity, robot actions, execution feedback, and object or device state changes. Beyond asking what an agent remembers, WorldLines tests whether the agent can use pre-cutoff visible history for question answering and state-aware planning. Formally, each instance is represented as

$$x_i = (\mathcal{H}_{<c_i}, q_i, S_{c_i}, \mathcal{E}_i, y_i^*, \tau_i),$$

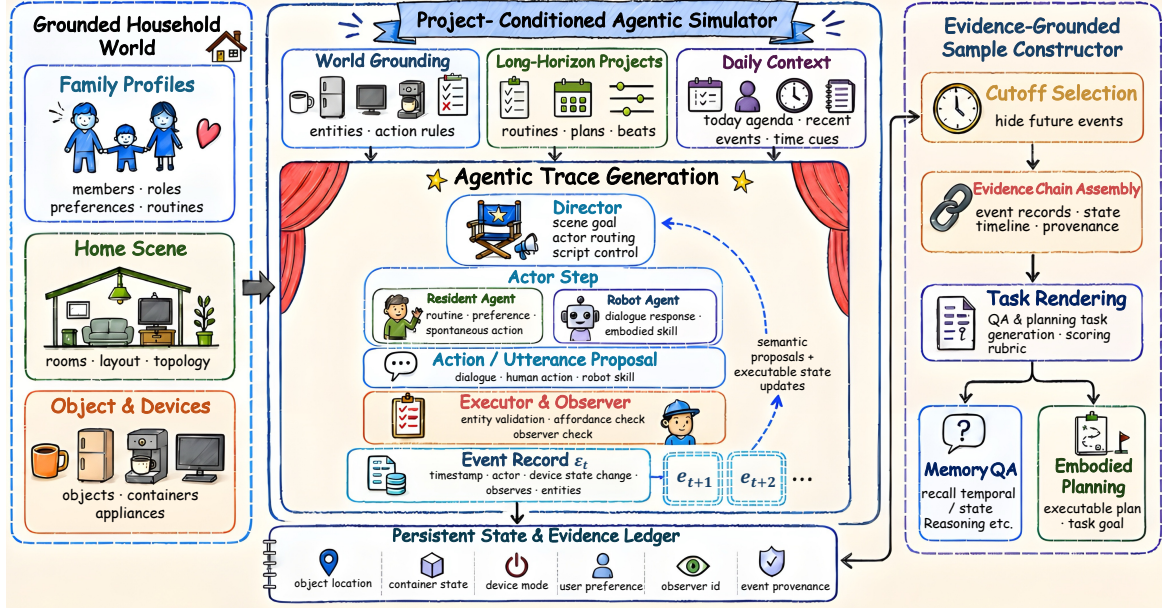


Figure 3: **Overview of the WorldLines construction framework.** WorldLines builds long-horizon embodied traces from grounded household worlds, project-driven activities, and closed-loop state-changing interactions. These histories are converted into cutoff-controlled, evidence-linked samples for *Memory QA* and *Embodied Task Planning*, testing persistent world-state maintenance under partial observability.

where $\mathcal{H}_{<c_i}$ is the visible history before cutoff c_i , q_i is the question or task instruction, S_{c_i} is the ground-truth world state, $\mathcal{E}_i$ is the supporting evidence chain, y_i^* is the reference answer or plan, and τ_i denotes the task type. The agent observes only $(\mathcal{H}_{<c_i}, q_i)$; states and evidence are used only for evaluation.

WorldLines contains two task families (Figure 2). *Memory QA* evaluates recovery of historical events, state changes, preferences, and routines. *Embodied Task Planning* evaluates whether an agent can generate a state-consistent plan or next-step decision from visible history.

3.2 Project-Driven Trace Generation

World grounding. Figure 3 summarizes the WorldLines construction pipeline. World grounding defines what exists in each household and what can be executed. WorldLines uses Habitat/HSSD household scenes as grounded environments and curates interactive objects, receptacles, and controllable devices. These assets are converted into a semantic view for project generation and an executable scene view for validation. The executable view organizes object instances, device states, openable components, and action constraints in a scene-graph-style representation, allowing generated actions to be affordance-checked and written into replayable state trajectories.

Project planning. Project planning gives each trace a persistent household thread. Instead of generating isolated daily events, WorldLines first creates long-term projects from the semantic view. Each project describes a multi-day life process, such as routine support, meal preparation, home organization, or device coordination, and specifies participants, relevant spaces and entities, temporal preferences, and constraints.

Closed-loop trace simulation. Closed-loop simulation turns long-term projects into cross-day event streams that change the world state. For day d , the generation context is

$$C_d = (\mathcal{W}, \mathcal{P}_d, S_{d-1}, N_{<d}),$$

where $\mathcal{W}$ is the grounded world, $\mathcal{P}_d$ denotes active projects, S_{d-1} is the accumulated state, and $N_{<d}$ contains carry-forward notes. Conditioned on C_d , the generator proposes dialogue, human activity, robot actions, and device operations. A deterministic executor checks entity references, affordances, and preconditions, and only valid outcomes are written as structured state changes.

Carry-forward memory. Carry-forward memory makes traces continuous across days. After each day, WorldLines extracts notes from events, execution feedback, and state changes, including changed object or device states, user preferences, unresolved plans, recent events, and potential con-

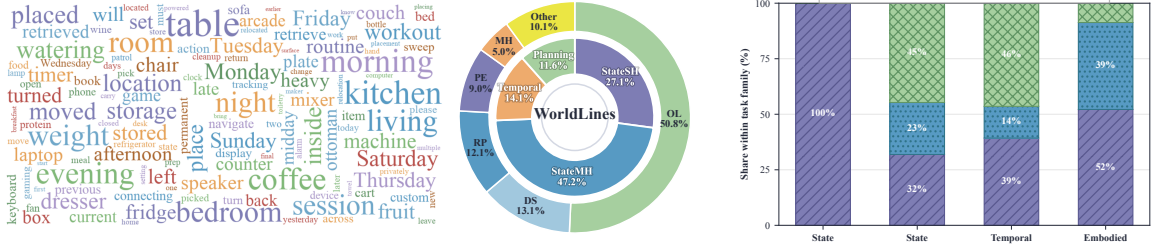


Figure 4: **Statistical overview of WorldLines.** (Left) Word distribution in benchmark queries and evidence. (Middle) sample distribution by task family (inner ring) and memory target (outer ring), including State Single-Hop (StateSH), State Multi-Hop (StateMH), Object Location (OL), Device State (DS), Routine Pattern (RP), Planning Execution (PE), and Multi-Hop Reasoning (MH). (Right) Temporal scope within each task family.

flicts. These notes are passed to later days with the accumulated state, so future interactions depend on earlier household history.

3.3 Evidence-Linked Task Construction

After generating cross-day traces, WorldLines constructs evaluation samples from events and state changes that actually occurred. It indexes events, state timelines, entity histories, and action histories, then mines candidate questions and planning tasks. The LLM only rewrites structured candidates into natural language; the cutoff, ground truth, and evidence chain are determined programmatically.

Each sample has a context cutoff. The evaluated agent observes only pre-cutoff history, and the reference answer or plan must be supported by pre-cutoff evidence. This prevents future information leakage while enabling event-level retrieval and state-consistency evaluation.

Memory QA covers current states, overwritten states, multi-hop temporal reasoning, preferences, routines, and source-aware questions. Embodied Task Planning requires a plan or next-step decision from the current instruction, historical state, and executable constraints. Together, these tasks test whether long-horizon state maintenance supports reliable QA and embodied decision making.

4 ObsMem: Observer-Grounded Memory

WorldLines requires memory systems to ground answers in past evidence, maintain mutable world states, and reason about what the robot is justified to know under partial observability. ObsMem addresses these requirements with four typed tracks for events, states, beliefs, and commitments.

4.1 Memory State Formulation

ObsMem treats the embodied history as a temporally ordered observation stream

$$\mathcal{O}_{1:T} = \{o_t\}_{t=1}^T, \quad o_t = (x_t, \tau_t, \ell_t, A_t, V_t),$$

where x_t is the event content, τ_t the timestamp, ℓ_t the location anchor, A_t the involved actors or entities, and V_t the observer set. Since the world is partially observable, ObsMem only ingests events visible to the robot, or information explicitly reported through visible dialogue.

Let r denote the robot. The observer gate stores only robot-visible events and creates a reported atom for visible utterances by other actors:

$$\Phi(o_t) = \begin{cases} \{e_t^{\text{obs}}\} \cup \mathbf{1}[\text{utt}(o_t) \wedge \text{actor}(o_t) \neq r] \{e_t^{\text{rep}}\}, & r \in V_t, \\ \emptyset, & r \notin V_t. \end{cases} \quad (1)$$

At time t , ObsMem maintains a typed memory state

$$\mathcal{M}_t = (\mathcal{E}_t, \mathcal{S}_t, \mathcal{B}_t, \mathcal{K}_t),$$

where $\mathcal{E}_t$ stores immutable event evidence, $\mathcal{S}_t$ stores structured state trails, $\mathcal{B}_t$ stores epistemic beliefs, and $\mathcal{K}_t$ stores future-oriented commitments such as requests, reminders, and promises.

4.2 Observer-Grounded Memory Update

ObsMem updates memory online through a visibility-gated loop: hidden events are never written into the agent memory. For each visible event, ObsMem appends an event atom to $\mathcal{E}_t$ and marks its provenance as observed or reported, preventing direct observations, reported claims, and unobserved changes from being collapsed into the same text memory.

For executable actions, ObsMem converts action schemas into structured state updates:

$$\Delta_t = (i, a, v_{\text{old}}, v_{\text{new}}, \tau_t, s_t),$$

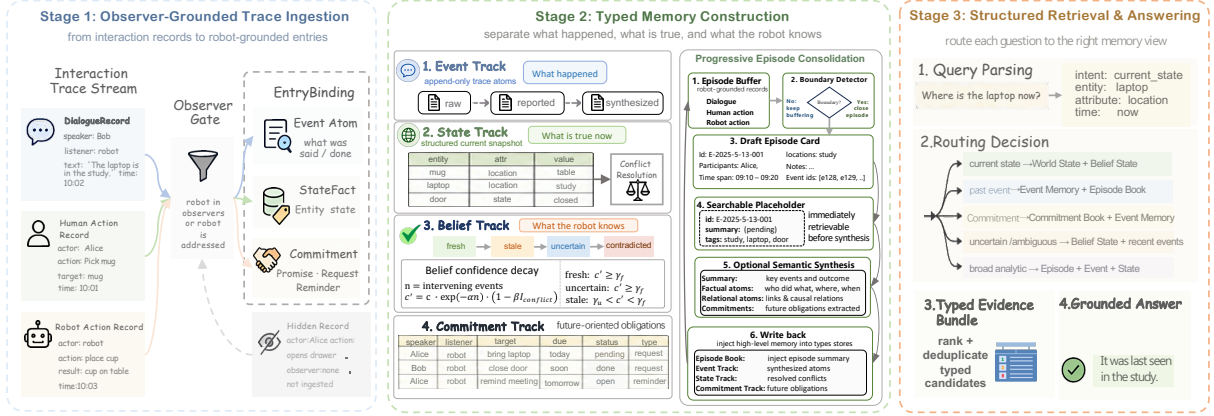


Figure 5: **Overview of ObsMem.** ObsMem organizes observer-grounded traces into typed views of events, states, beliefs, and commitments, then consolidates episodes for structured retrieval and grounded answering.

where entity i has attribute a changed from v_{old} to v_{new} at time τ_t with source s_t . These updates are written into $\mathcal{S}_t$, which maintains both the latest snapshot and historical state trail, enabling current-state and overwritten-state reasoning.

Each state update also refreshes the belief track. Direct observations receive high confidence, while reported information receives lower confidence. Later intervening events by other actors can decay a belief from *fresh* to *stale* or *uncertain*, and conflicting evidence marks it as *contradicted*. Thus, ObsMem records both the last supported state and whether the robot is still justified in believing it.

Let I_t be intervening events affecting (i, a) , A_t their actors, and C_t contradicting evidence. With fixed thresholds λ_I , λ_m , and λ_A , the epistemic state is updated as

$$z_t(i, a) = \begin{cases} \text{contradicted,} & |C_t| > 0, \\ \text{uncertain,} & |I_t| \geq \lambda_I \vee (|I_t| \geq \lambda_m \wedge |A_t| \geq \lambda_A), \\ \text{stale,} & |I_t| > 0 \vee \rho_t \neq \text{observed}, \\ \text{fresh,} & \text{otherwise.} \end{cases} \quad (2)$$

At episode boundaries, ObsMem performs progressive consolidation over buffered visible events. It converts local event windows into searchable episode cards and writes high-level facts, relations, and commitments back to the typed stores. If consolidation fails, low-level event atoms and state trails are still preserved.

4.3 Typed Retrieval and Reasoning

Given a question or task instruction q , ObsMem retrieves evidence through typed routing rather than a single flat memory search. It first predicts a query plan

$$p_q = (\eta_q, I_q, A_q, T_q),$$

where η_q is the intent, I_q are target entities, A_q are relevant state attributes, and T_q is an optional temporal constraint. The plan routes retrieval to the appropriate views: current-state queries prioritize $\mathcal{S}_t$, past-event queries retrieve $\mathcal{E}_t$ and episode cards, epistemic queries access $\mathcal{B}_t$, and future-oriented queries may use commitments.

For answering and planning, ObsMem combines event evidence, state facts, and belief records into a grounded context. For example, a question asking where a cup is now is routed mainly to state trails, while asking whether the robot knows its location also retrieves belief records. For planning, this context exposes both relevant states and uncertainty signals, helping avoid actions based on stale or unobserved world states.

5 Experiment

5.1 Experiment Setup

Figure 4 summarizes the scale and temporal coverage of WorldLines. Beyond raw trace size, WorldLines emphasizes delayed memory use and cross-day evidence, requiring systems to recover precise state-changing events rather than only broadly relevant sessions.

We evaluate Mem0 (Chhikara et al., 2025), Amem (Xu et al., 2026b), MemoryOS (Kang et al., 2025), GraphMem (Hu et al., 2026a), Graphiti (He et al., 2025), HippoRAG2 (Gutiérrez et al., 2025), and our proposed ObsMem on two evidence-linked tasks: *Memory QA* and *Embodied Task Planning*. All systems receive the same cutoff-controlled visible history. For QA, each system passes at most five retrieved records to the answer generator, and retrieval metrics are computed over these top-five

Method	Judge $\uparrow$	Perfect $\uparrow$	Sess. Any@5 $\uparrow$	Event R@5 $\uparrow$	StateMH-J $\uparrow$	StateMH-E $\uparrow$	StateSH-J $\uparrow$	Temp-J $\uparrow$
A-mem	0.575	53.2%	0.839	0.355	0.540	0.216	0.550	0.692
Mem0	0.554	53.2%	0.823	0.378	0.598	0.264	0.550	0.462
GraphMem	0.457	38.7%	0.806	0.243	0.529	0.184	0.417	0.359
MemoryOS	0.312	29.0%	0.452	0.085	0.287	0.086	0.350	0.308
Graphiti	0.642	58.1%	0.806	0.331	0.356	0.158	0.383	0.627
HippoRAG2	0.613	53.2%	0.774	0.274	0.621	0.172	0.600	0.615
ObsMem	0.713	69.0%	0.879	0.537	0.762	0.452	0.667	0.667

Table 2: **Memory QA performance on WorldLines.** All methods are evaluated on 310 Memory QA samples. We report overall QA quality, session/event retrieval, and state- or temporal-reasoning diagnostics. StateMH-E highlights event-level recall in the most evidence-demanding multi-hop state setting. Dashes denote unavailable question-family results. Full family-level breakdowns are in Appendix A.8.

Variant	Ablated Component	Judge $\uparrow$	Δ Judge	Perfect $\uparrow$	Event R@5 $\uparrow$	Hidden Judge $\uparrow$	Latency $\downarrow$
w/ Full ObsMem	–	0.699	–	66%	0.563	0.278	8.82
w/o Belief	Belief-view retrieval	0.651	-0.048	63%	0.558	0.000	8.63
w/o State	World-state retrieval	0.597	-0.102	56%	0.532	0.111	8.95
w/o Consol.	Episode consolidation	0.554	-0.145	53%	0.419	0.167	9.16
w/o Selector	Evidence selector	0.435	-0.264	40%	0.466	0.000	5.54

Table 3: **ObsMem ablation results on a 62-sample diagnostic QA subset.** Each variant ablates one query-time memory view or mechanism while keeping the rest of the pipeline unchanged. Hidden Judge is computed on 6 hidden-until-observed questions. Latency is measured in seconds. Additional details are provided in Appendix A.5.

records. Questions and reference answers are generated with GPT-4o-mini under evidence-linked constraints and manually verified against annotated supporting evidence; see Appendix A.2. All memory systems use google/gemini-3.5-flash for answer generation, and GPT-4o serves as the independent judge. Additional baseline and evaluation details are provided in Appendix A.1.

Metrics. We report **Judge**, the normalized LLM-as-a-judge answer-quality score; **Perfect Rate**, the fraction of answers with a normalized score of 1.0; **Session Any@5**, the fraction of queries with at least one gold supporting session among the top-five retrieved records; and **Event R@5**, the average fraction of gold evidence events covered by the top-five records. Session-level recall measures coarse context coverage, while event-level recall evaluates precise embodied evidence recovery. Planning metrics are defined in Appendix A.6. Efficiency and token-cost statistics are reported as supplementary analysis in Appendix A.7.

Judge reliability. To validate the automatic judge, we sample 80 system outputs covering all question families and evaluated methods. Human labels show substantial agreement (Fleiss’ $\kappa = 0.71$), and GPT-4o reaches 87.5% agreement with the majority human label and 0.82 Spearman correlation with averaged human scores. Judge validation and multi-backbone results are reported in Appendix A.3.

5.2 Overall Evaluation

Table 2 shows that ObsMem achieves the strongest overall Memory QA performance. It obtains the highest Judge score and Perfect Rate, indicating that observer-grounded memory improves downstream answer quality. Compared with the strongest baseline on each metric, ObsMem improves Judge by 0.071 over Graphiti and Event R@5 by 0.159 over Mem0. This suggests that its typed state trails and event-grounded retrieval help recover the concrete household evidence needed for answering.

The results also reveal a key property of WorldLines: retrieving a broadly relevant session is not sufficient for embodied memory. Several baselines obtain relatively high Session Any@5, showing that they often reach the correct coarse temporal context, while their Event R@5 remains substantially lower than ObsMem. In dynamic household environments, a session may contain many object movements, device operations, and dialogue reports. Correct answering therefore requires identifying the exact state-changing event, not only retrieving semantically related text.

We add FullContext, Grep+LLM, and Oracle Evidence controls. Grep+LLM uses Gemini Flash to extract 2–3 query keywords (at most 5), retrieves up to 15 case-insensitive matching events, and passes them to the same Gemini Flash answer model.

Case Study

Day 1 evening Bob’s weekend coffee routine is set to 08:30.
 Day 3 night Exhausted after work, Bob mistakenly sets the kitchen coffee timer to 04:00.
 Day 4 early morning The robot detects the abnormal timer and corrects it to Bob’s weekday routine, 07:00.
 Query “What time did Bob mistakenly set the coffee timer to, and what time did the robot correct it to?”

▲ Flat-text memory (Mem0)

Retrieves the salient weekend routine 08:30 and the later correction to 07:00.

Answer: “Bob set it to 08:30, and the robot corrected it to 07:00.”

Failure: confuses a recurring routine with a one-off erroneous state.

✓ ObsMem

Maintains a typed state trail for the coffee timer: routine, anomalous update, and corrective update.

Answer: “Bob mistakenly set it to 04:00, and the robot corrected it to 07:00.”

Why: separates routine=08:30, anomaly=04:00, and correction=07:00 by entity, time, and actor.

Method	Judge ↑	Perfect ↑	Sess. Any@5 ↑	Event R@5 ↑
FullContext	0.758	74.2%	—	—
ObsMem	0.713	69.0%	0.879	0.537
Grep+LLM	0.349	16.0%	0.306	0.165
Oracle Evidence	0.952	90.3%	1.000	1.000

Table 4: **Context and oracle controls on Memory QA.** All methods use Gemini Flash for answer generation.

The question-family columns further show that ObsMem performs best on StateMultiHop and StateSingleHop questions, which require tracking mutable household states through one or more event transitions. We include StateMultiHop Event R@5 in the main table because this family requires recovering multiple state-changing events rather than only locating a broadly relevant session. Figure 6 illustrates this failure mode: a flat-text memory retrieves the salient routine but misses the anomalous state update, while ObsMem preserves the full state trail. The StateMH-E column shows a particularly large gap on event grounding for multi-hop state questions, where ObsMem improves over the strongest baseline by 0.188. A-mem remains competitive on TemporalMemory questions, suggesting that text-centric memories can be effective when temporal cues are explicit. Detailed event-level breakdowns and radar visualizations are provided in Appendix A.8 and Appendix A.4.

5.3 Ablation Study

We conduct ablations on a 62-sample diagnostic subset covering the three question families and the main ObsMem mechanisms. The full ObsMem row is re-evaluated on this subset, so the results are not directly comparable to Table 2. Table 3 ablates four components: belief-view retrieval, world-state retrieval, episode consolidation, and LLM-based evidence selection. The w/o Belief and w/o State variants remove only query-time access to the cor-

Method	Plan ↑	State ↑	Precond. ↑	Mem. ↑
A-mem	0.542	0.566	0.581	0.524
Mem0	0.526	0.551	0.563	0.512
GraphMem	0.481	0.493	0.507	0.462
MemoryOS	0.337	0.361	0.376	0.319
ObsMem	0.684	0.721	0.702	0.690

Table 5: **Downstream embodied planning results.** Full planning dimensions are provided in Appendix A.6.

responding memory views while keeping ingestion unchanged.

The evidence selector is the most critical component: removing it drops Judge from 0.699 to 0.435, despite moderate Event R@5, showing that answer quality depends on selecting and combining evidence across memory views, not only retrieval. Disabling episode consolidation causes the second largest drop, reducing Judge to 0.554 and Event R@5 to 0.419, indicating the value of episode synthesis and conflict resolution. Removing world-state retrieval hurts current-state reasoning, since object locations and device states must be reconstructed from raw events. The w/o Belief variant has a smaller overall impact, but reduces Hidden Judge from 0.278 to 0.000, suggesting that episodic belief tracking is especially useful under partial observability.

5.4 Downstream Embodied Planning Evaluation

Beyond Memory QA, we evaluate whether retrieved long-term memory can support executable, state-aware household planning. This setting is more demanding than QA because an agent must not only recall relevant evidence, but also use it to check object locations, container states, device states, and action preconditions before producing an action sequence. Although the planning set contains 21 samples, each instance is action-dense,

with an average of 7.6 target actions and 3.1 remembered state constraints.

Table 5 reports planning results. ObsMem obtains the highest Plan Judge score and performs especially well on state consistency, precondition validity, and memory use. These gains suggest that typed state trails help translate remembered household states into executable plans. In contrast, text-centric or graph-expanded memory systems can retrieve useful task context but are less reliable at converting retrieved information into explicit state constraints. Full planning dimensions are provided in Appendix A.6.

6 Conclusion

We introduced **WorldLines**, a benchmark for evaluating long-horizon stateful embodied agents in dynamic and partially observable household environments. Unlike prior memory or embodied-task benchmarks, WorldLines focuses on whether agents can maintain persistent world states across dialogue, human activity, robot actions, device changes, and execution feedback, and use them for Memory QA and Embodied Task Planning. We further proposed **ObsMem**, an observer-grounded memory framework that separates event evidence, structured world states, and agent beliefs to support state-aware reasoning under partial observability. Experiments show that existing memory systems struggle with overwritten states, uncertainty, and translating long-term memory into embodied decisions, highlighting the need for memory architectures designed for stateful embodied interaction.

7 Limitations

WorldLines is constructed in simulated household environments and does not fully cover perception noise, actuation errors, or open-ended human behavior in real homes. This controlled setting enables precise annotation of evidence chains, cutoffs, state changes, and executable constraints for systematic evaluation. Future work can extend WorldLines to real robot logs, visual observations, and full physical simulation.

ObsMem is designed for structured embodied traces with entity identifiers, visibility annotations, and action schemas. In real deployment, these signals would need to be provided by perception, localization, and grounding modules. ObsMem also introduces additional latency from typed retrieval and belief-aware evidence selection, motivating

more efficient retrieval and integration with visual perception and execution feedback.

References

- Haonan Bian, Zhiyuan Yao, Sen Hu, Zishan Xu, Shaolei Zhang, Yifu Guo, Ziliang Yang, Xueran Han, Huacan Wang, and Ronghao Chen. 2026. Realmem: Benchmarking llms in real-world memory-driven interaction. *arXiv preprint arXiv:2601.06966*.
- Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Motlaghi, Priyam Parashar, and 1 others. 2025. Partnr: A benchmark for planning and reasoning in embodied multi-agent tasks. In *International Conference on Learning Representations*, volume 2025, pages 65205–65268.
- Ding Chen, Simin Niu, Kehang Li, Peng Liu, Xiangping Zheng, Bo Tang, Xinchu Li, Feiyu Xiong, and Zhiyu Li. 2026. Halumem: Evaluating hallucinations in memory systems of agents. *Preprint*, arXiv:2511.03506.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.
- Meng Chu, Xuan Billy Zhang, Kevin Qinghong Lin, Lingdong Kong, Jize Zhang, Teng Tu, Weijian Ma, Ziqi Huang, Senqiao Yang, Wei Huang, Yeying Jin, Zhefan Rao, Jinhui Ye, Xinyu Lin, Xichen Zhang, Qisheng Hu, Shuai Yang, Leyang Shen, Wei Chow, and 23 others. 2026. Agentic world modeling: Foundations, capabilities, laws, and beyond. *Preprint*, arXiv:2604.22748.
- Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Motlaghi. 2022. Proctor: Large-scale embodied ai using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994.
- Pascale Fung, Yoram Bachrach, Asli Celikyilmaz, Kamalika Chaudhuri, Delong Chen, Willy Chung, Emmanuel Dupoux, Hongyu Gong, Hervé Jégou, Alessandro Lazaric, Arjun Majumdar, Andrea Madotto, Franziska Meier, Florian Metze, Louis-Philippe Morency, Théo Moutakanni, Juan Pino, Basile Terver, Joseph Tighe, and 2 others. 2025. Embodied ai agents: Modeling the world. *Preprint*, arXiv:2506.22355.
- Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, and 1 others. 2024. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028. IEEE.
- Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. *Preprint*, arXiv:2502.14802.
- Yang He, Ruijie Fang, Işıl Dillig, and Yuepeng Wang. 2025. Graphiti: Bridging graph and relational database queries. *Proc. ACM Program. Lang.*, 9(PLDI).
- Sen Hu, Yuxiang Wei, Jiaxin Ran, Zhiyuan Yao, Xueran Han, Huacan Wang, Ronghao Chen, and Lei Zou. 2026a. Does memory need graphs? a unified framework and empirical analysis for long-term dialog memory. *Preprint*, arXiv:2601.01280.
- Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxiang Sun, Yutao Zhu, Hao Sun, Boci Peng, and 28 others. 2026b. Memory in the age of ai agents. *Preprint*, arXiv:2512.13564.
- Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. 2025. Memory OS of AI agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25961–25970, Suzhou, China. Association for Computational Linguistics.
- Taeyoon Kwon, Dongwook Choi, Hyojun Kim, Sunghwan Kim, Seungjun Moon, Beong-woo Kwak, Kuan-Hao Huang, and Jinyoung Yeo. 2025. Embodied agents meet personalization: Investigating challenges and solutions through the lens of memory utilization. *arXiv preprint arXiv:2505.16348*.
- Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabriel Levine, Michael Lingelbach, Jiankai Sun, and 1 others. 2023. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR.
- Hansen Jin Lillemark, Benhao Huang, Fangneng Zhan, Yilun Du, and Thomas Anderson Keller. 2026. Flow equivariant world models: Memory for partially observed dynamic environments. *Preprint*, arXiv:2601.01075.
- Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870.
- Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. Memgpt: Towards llms as operating systems. *Preprint*, arXiv:2310.08560.
- Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander Clegg, Michal Hlavac, So Yeon Min, and 1 others. 2024. Habitat 3.0: A co-habitat for humans, avatars, and robots. In *International Conference on Learning Representations*, volume 2024, pages 15306–15336.

- Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. 2023. Say-plan: Grounding large language models using 3d scene graphs for scalable task planning. In *7th Annual Conference on Robot Learning*.
- Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. [{ALFW}orld: Aligning text and embodied environments for interactive learning](#). In *International Conference on Learning Representations*.
- Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2024. [Voyager: An open-ended embodied agent with large language models](#). *Transactions on Machine Learning Research*.
- Zixuan Wang, Bo Yu, Junzhe Zhao, Wenhao Sun, Sai Hou, Shuai Liang, Xing Hu, Yinhe Han, and Yiming Gan. 2025. Karma: Augmenting embodied ai agents with long-and-short term memory systems. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE.
- Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. [Longmemeval: Benchmarking chat assistants on long-term interactive memory](#). In *The Thirteenth International Conference on Learning Representations*.
- Quanting Xie, So Yeon Min, Pengliang Ji, Yue Yang, Tianyi Zhang, Kedi Xu, Aarav Bajaj, Ruslan Salakhutdinov, Matthew Johnson-Roberson, and Yonatan Bisk. 2025. [Embodied-rag: General non-parametric embodied memory for retrieval and generation](#). *Preprint*, arXiv:2409.18313.
- Buqiang Xu, Yijun Chen, Jizhan Fang, Ruobin Zhong, Yunzhi Yao, Yuqi Zhu, Lun Du, and Shumin Deng. 2026a. [Structmem: Structured memory for long-horizon behavior in llms](#). *Preprint*, arXiv:2604.21748.
- Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2026b. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604.
- Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. 2025. [Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents](#). In *Forty-second International Conference on Machine Learning*.
- Pukun Zhao, Longxiang Wang, Miaowei Wang, Chen Chen, Fanqing Zhou, and Haojian Huang. 2025. [Evoempirbench: Dynamic spatial reasoning with agent-expver](#). *Preprint*, arXiv:2509.12718.
- Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. [Memorybank: enhancing large language models with long-term memory](#). In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence, AAAI’24/IAAI’24/EAAI’24*. AAAI Press.
- Qinhong Zhou, Sunli Chen, Yisong Wang, Haozhe Xu, Weihua Du, Hongxin Zhang, Yilun Du, Joshua B. Tenenbaum, and Chuang Gan. 2024. [HAZARD challenge: Embodied decision making in dynamically changing environments](#). In *The Twelfth International Conference on Learning Representations*.

A Additional Experimental Details

A.1 Baseline Implementation Details

We evaluate Mem0, A-mem, MemoryOS, and GraphMem as representative long-term memory baselines. When an official implementation is available, we use the official codebase and retain the default memory-update procedure recommended by the original method. When a method requires a system-specific memory construction procedure, we keep that procedure unchanged because memory updating is part of the method being evaluated. All methods receive the same cutoff-controlled visible household history before each query.

For answer generation, we enforce a shared context budget across systems. Each method may retrieve using its own internal scoring or expansion mechanism, but only the top five final retrieved records are passed to the answer generator. Retrieval metrics are computed over the same top-five records. For GraphMem, graph expansion is allowed during retrieval, but the final graph-expanded context is capped to five retrieved records before answer generation. This ensures that downstream answer quality is not driven by unequal context length.

ObsMem differs from generic text-memory baselines by using observer-grounded storage, typed state trails, and belief-aware retrieval. Its retriever may return fewer than five records when fewer high-confidence typed evidence records are available.

A.2 QA Verification Protocol

Benchmark questions and reference answers are generated under evidence-linked constraints and then manually verified against annotated supporting evidence. During verification, annotators check that each QA item satisfies three criteria: (1) the question is answerable before the context cutoff, (2) the reference answer is entailed by annotated supporting evidence, and (3) answering the question does not require post-cutoff information or evaluator-only hidden state. Items failing any criterion are revised or removed.

A.3 Judge Validation Protocol

We validate GPT-4o on 80 outputs spanning all methods and question families. Five annotators use the same 0–3 rubric. Human agreement reaches Fleiss’ $\kappa = 0.71$; GPT-4o achieves 87.5% majority-label agreement and 0.82 Spearman correlation

with mean human scores. Session Any@5 and Event R@5 remain judge-independent.

Backbone robustness. We evaluate ObsMem with three backbones on the same 62-sample diagnostic subset. The results quantify model sensitivity while retaining substantial evidence retrieval across backbones.

A.4 Radar Visualization by Embodied Memory Type

Figure 7 provides a detailed visualization of method behavior across embodied memory types. The contrast between Session Any@5 and Event R@5 shows that methods can retrieve broadly relevant sessions without recovering the exact state-changing events required by the answer.

A.5 Ablation Variant Details

The ablation study removes one central ObsMem mechanism at a time while keeping the rest of the ingestion, retrieval, answer-generation, and judging pipeline unchanged. We run the ablation on a 62-sample diagnostic QA subset covering State-MultiHop, StateSingleHop, and TemporalMemory questions.

A.6 Downstream Embodied Planning Probe

We score plans along four dimensions. *State consistency* checks whether the plan respects remembered object, container, and device states. *Precondition validity* checks whether each proposed action is executable under the current state. *Memory use* checks whether relevant long-term household context is incorporated. *Action order* checks whether the proposed steps form a coherent executable sequence. Plan Judge is the average of these four scores.

The planning results serve as a smaller-sample downstream planning probe. Each planning instance contains an average of 7.6 target actions and requires checking 3.1 remembered state constraints on average, including object locations, container states, device settings, and action preconditions. On this action-dense set, ObsMem shows promising gains across all dimensions, especially state consistency and precondition validity. This suggests that structured state trails may help generate more state-aware plans.

Method	Implementation	Memory Update	Retrieval / Context Cap
Mem0	Official	Default extraction/update pipeline	Top-5 final records
A-mem	Official	Agentic memory evolution	Top-5 final records
MemoryOS	Official	Hierarchical memory update	Top-5 final records
GraphMem	Reimplemented from specification	Graph construction and expansion	Top-5 graph-expanded records
ObsMem	Ours	Observer-grounded state-trail update	Up to 5 typed evidence records

Table 6: **Baseline implementation and context control.** All methods receive the same cutoff-controlled visible history. Retrieval can use method-specific internal mechanisms, but the final answer-generation context is capped to at most five records.

Score	Meaning
0	Incorrect or contradicts reference evidence
1	Partially relevant but misses key evidence
2	Mostly correct with minor omissions
3	Fully correct and evidence-consistent

Table 7: **Judge rubric for Memory QA answer correctness.** The automatic judge and human annotators use the same 0–3 rubric.

A.7 Full Efficiency and Context-Cost Statistics

All systems use the same maximum answer-generation budget of five retrieved records. ObsMem often returns fewer records because its typed retriever abstains from adding weakly matched evidence. GraphMem uses the largest prompt context: although it is capped to five final retrieved records, each graph-expanded record may contain neighboring node descriptions, resulting in a larger prompt context.

A.8 Additional Metrics

We report additional question-family event-level metrics in Table 13. These diagnostics complement the compact main-table results and further show the gap between coarse session retrieval and precise event-level grounding.

B Replay and Edited Scene Examples

WorldLines is grounded in Habitat/HSSD household scenes rather than text-only interaction logs. For each family, we manually curate the base scene by selecting interaction-relevant objects, valid receptacles, controllable devices, and household roles. The generated event traces can then be replayed against the edited scene state. These visualizations are used for qualitative inspection and illustration; evaluation itself uses the structured event logs, state transitions, cutoff annotations, and

evidence links described in the main paper. Figure 8 provides replay examples, and Figure 9 shows a placeholder for the edited scene interface screenshot.

C Benchmark Taxonomy and Action Space

WorldLines uses a controlled vocabulary to generate long-horizon household traces. It consists of three parts: project types, memory targets, and executable skills.

Project Types. Project types define the high-level household themes used to organize long-horizon traces:

- **household_routine:** recurring activities that structure daily household life.
- **routine_support:** support for an individual’s repeated habits, schedules, or preparation needs.
- **household_organization:** ongoing organization, storage, tidying, or rearrangement themes.
- **health_lifestyle:** health, exercise, recovery, or lifestyle-related household projects.
- **meal_preparation:** food, drink, grocery, meal planning, or kitchen-related routines.
- **work_study_support:** work, study, documents, workspace, or deadline-driven support.
- **digital_device_coordination:** smart-device, remote-control, or digital state coordination.
- **family_comfort:** comfort, relaxation, entertainment, or family well-being activities.

Memory Targets. Memory targets specify the intended memory challenges within each project:

Base Model	Judge $\uparrow$	Perfect $\uparrow$	Sess. Any@5 $\uparrow$	Event R@5 $\uparrow$	StateMH-J $\uparrow$	StateSH-J $\uparrow$	Temp-J $\uparrow$
Gemini Flash	0.667	58.1%	0.839	0.460	0.667	0.667	0.667
Qwen3.5-Plus	0.543	46.8%	0.839	0.410	0.517	0.583	0.538
DeepSeek-V4	0.565	50.0%	0.887	0.468	0.437	0.667	0.692

Table 8: **ObsMem robustness across LLM backbones.** All models are evaluated on the same 62-sample diagnostic subset with a fixed GPT-4o judge.

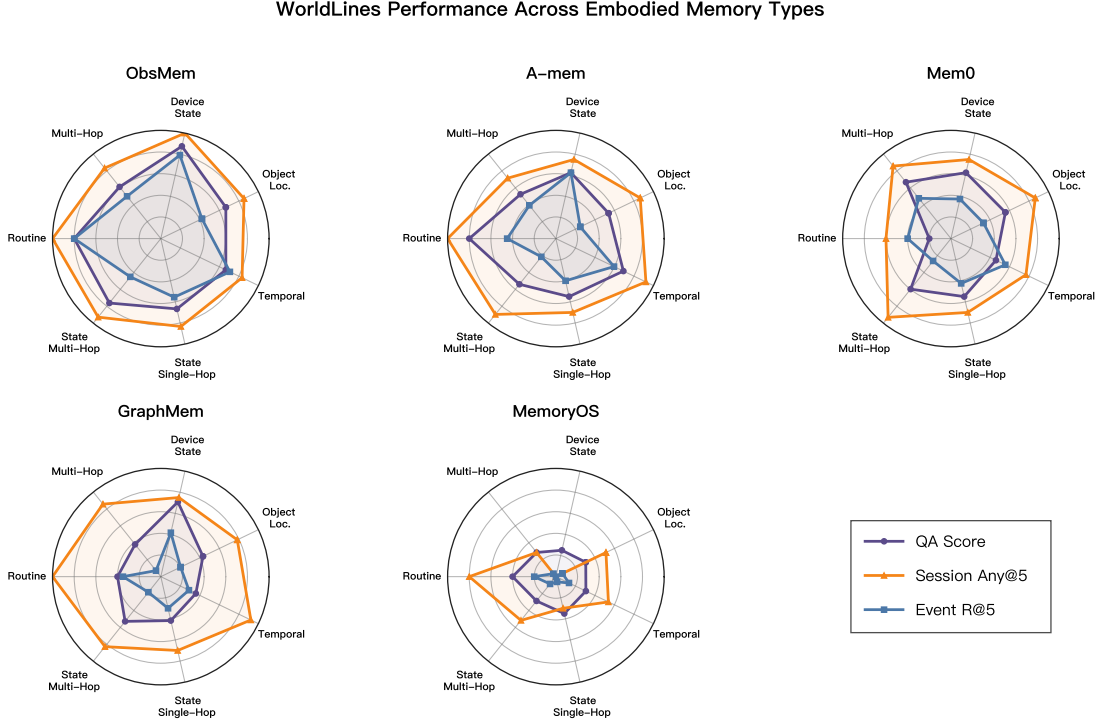


Figure 7: **Performance across embodied memory question types.** Each radar chart corresponds to one memory system. Axes denote representative embodied memory categories, and curves report QA score, session recall, and event recall. ObsMem shows a more balanced profile across state-centric categories, while generic memory systems often retrieve coarse sessions without matching the same level of event grounding.

Subset Type	N
StateMultiHop	29
StateSingleHop	20
TemporalMemory	13
Hidden-until-observed	6

Table 9: **Ablation diagnostic subset composition.** Question-family counts sum to 62. Hidden-until-observed is a visibility subset used for the Hidden Judge diagnostic.

- preference: user preferences revealed through dialogue, routines, or repeated choices.
- routine: repeated behavior patterns, habits, or “usual” household arrangements.
- planning_dependency: facts that affect later planning, task ordering, or delayed execution.
- hidden_state: state changes that are not directly observed by the robot or another actor.
- social_context: social commitments, interpersonal context, or family coordination needs.
- object_location: where movable objects are placed, moved, hidden, or retrieved.
- temporal_state: time-sensitive commitments, deadlines, schedules, and recent changes.
- device_state: functional states of controllable devices, such as power, mode, or timer.

Action Space. The action space defines the executable skill interface used by the closed-loop trace generator:

- navigate_to: move an actor to a room, furniture, device, or object anchor before physical

Variant	Removed Component	Implementation Change
NoBelief	Belief retrieval	Belief records are still maintained during ingestion, but the router does not surface belief candidates to the answerer.
NoState	World-state retrieval	World-state facts are still maintained for diagnostics, but state candidates are removed from query-time retrieval.
NoConsol	Episode consolidation	Episode synthesis and state-conflict resolution are disabled; only minimal structural placeholders are retained for interface compatibility and contain no additional semantic summaries.
NoSelector	Evidence selector	The LLM reranker is bypassed, and candidates are passed in router-emitted order.

Table 10: **ObsMem ablation variants.** Each variant isolates one mechanism in the ObsMem memory pipeline without changing the benchmark samples, backbone model, judge, or answer-generation prompt format.

Method	N	Plan Judge $\uparrow$	State Consistency $\uparrow$	Precondition Validity $\uparrow$	Memory Use $\uparrow$	Action Order $\uparrow$
ObsMem	21	0.684	0.721	0.702	0.690	0.676
A-mem	21	0.542	0.566	0.581	0.524	0.553
Mem0	21	0.526	0.551	0.563	0.512	0.535
GraphMem	21	0.481	0.493	0.507	0.462	0.496
MemoryOS	21	0.337	0.361	0.376	0.319	0.352

Table 11: **Downstream planning probe on 21 action-dense samples.** Planning requires agents to use remembered household states for executable, state-aware action decisions.

interaction.

- inspect: observe a target object, receptacle, device, or furniture item to reveal state or contextual information.
- pick: pick up a movable object or device node, subject to co-location and empty-hand constraints.
- place: place the held object onto or into a valid destination surface, receptacle, furniture, or device.
- open: open an openable container, furniture item, or device component.
- close: close an openable container, furniture item, or device component.
- set_device_state: modify a supported smart-device state field, such as power, mode, temperature, or timer.
- handoff: transfer a held object from one co-located actor to another actor with empty hands.

D Prompt Templates

This appendix presents the key prompt templates used in our data generation pipeline and evaluation protocol. Placeholders in {curly braces} are filled programmatically at runtime.

D.1 Project Candidate Generation (Stage 2)

The following system prompt instructs the LLM to generate diverse candidate household life-theme projects for a given family and home environment. The LLM outputs semantic project descriptions without binding to concrete object IDs or action sequences; downstream stages ground each project into executable traces. Figure 10 shows the condensed prompt excerpt.

D.2 Project Day Beat Planning (Stage 3A)

Stage 3A expands each selected household project into a sparse multi-day semantic trajectory. It decides when a project is established, interrupted, recovered, or completed, while deliberately avoiding concrete sessions, rooms, actions, or state changes. Figure 11 shows the corresponding prompt excerpt.

D.3 Session Intent Planning (Stage 3B)

Stage 3B converts the project beats active on a given day into ordered household sessions. It plans life situations, timing, visibility, and narrative pressure, but still avoids executable actions and concrete object manipulation. Figure 12 shows the prompt excerpt used at this stage.

D.4 Closed-Loop Agentic Trace Generation

The final trace generator uses a closed-loop director–actor–executor protocol rather than a single scriptwriter. The director first creates a compact

Method	Avg. Latency (s) ↓	Prompt Tok. ↓	Completion Tok. ↓	Avg. Passed Records
A-mem	3.35	625	498	5.0
Mem0	4.57	513	539	5.0
MemoryOS	4.04	507	378	5.0
GraphMem	4.43	2057	388	5.0
ObsMem	5.13	562	319	3.4

Table 12: **Efficiency and context cost.** All systems pass at most five retrieved records to the answer generator. Avg. Passed Records measures final context size rather than retrieval correctness; lower values indicate fewer records passed to the generator, not necessarily better retrieval. ObsMem often returns fewer records due to typed retrieval, while incurring higher latency from structured state and belief-aware retrieval.

Method	StateMultiHop (145)		StateSingleHop (100)		TemporalMemory (65)	
	Judge ↑	Event R@5 ↑	Judge ↑	Event R@5 ↑	Judge ↑	Event R@5 ↑
ObsMem	0.762	0.452	0.667	0.556	0.667	0.708
A-mem	0.540	0.216	0.550	0.400	0.692	0.596
Mem0	0.598	0.264	0.550	0.425	0.462	0.558
GraphMem	0.529	0.184	0.417	0.300	0.359	0.288
MemoryOS	0.287	0.086	0.350	0.050	0.308	0.135

Table 13: **Full question-family performance with event-level grounding.** The main paper reports a compact subset of these family-level metrics; this appendix table provides the complete Event R@5 breakdown for methods with available question-family outputs.

session setup, actors then propose executable event sequences, and the deterministic executor applies state changes and derives visibility. Figures 13, 14, and 15 show concise prompt excerpts for the three LLM-facing components.

D.5 Evaluation and ObsMem Runtime Prompts

We also expose the two prompts most relevant to evaluation transparency and ObsMem reproducibility. Figure 16 shows the LLM-as-Judge rubric used for Memory QA scoring, while Figure 17 summarizes the ObsMem prompts that convert a question into typed retrieval views and select final evidence.

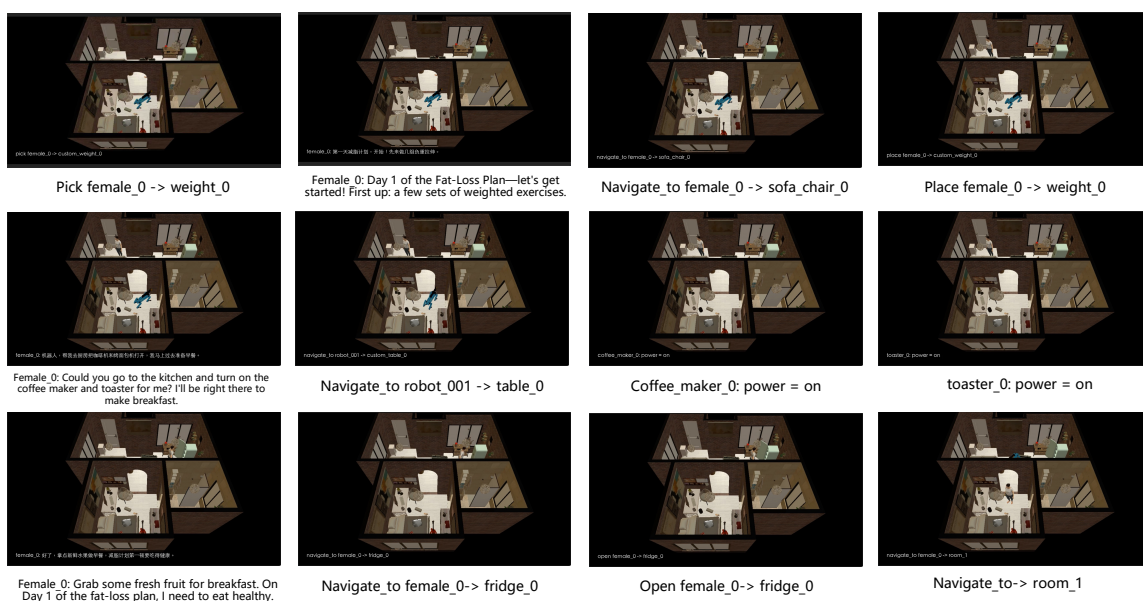


Figure 8: **Replay examples.** Generated household traces can be replayed in the corresponding edited Habitat scene to visualize robot & human npc actions, object relocation.

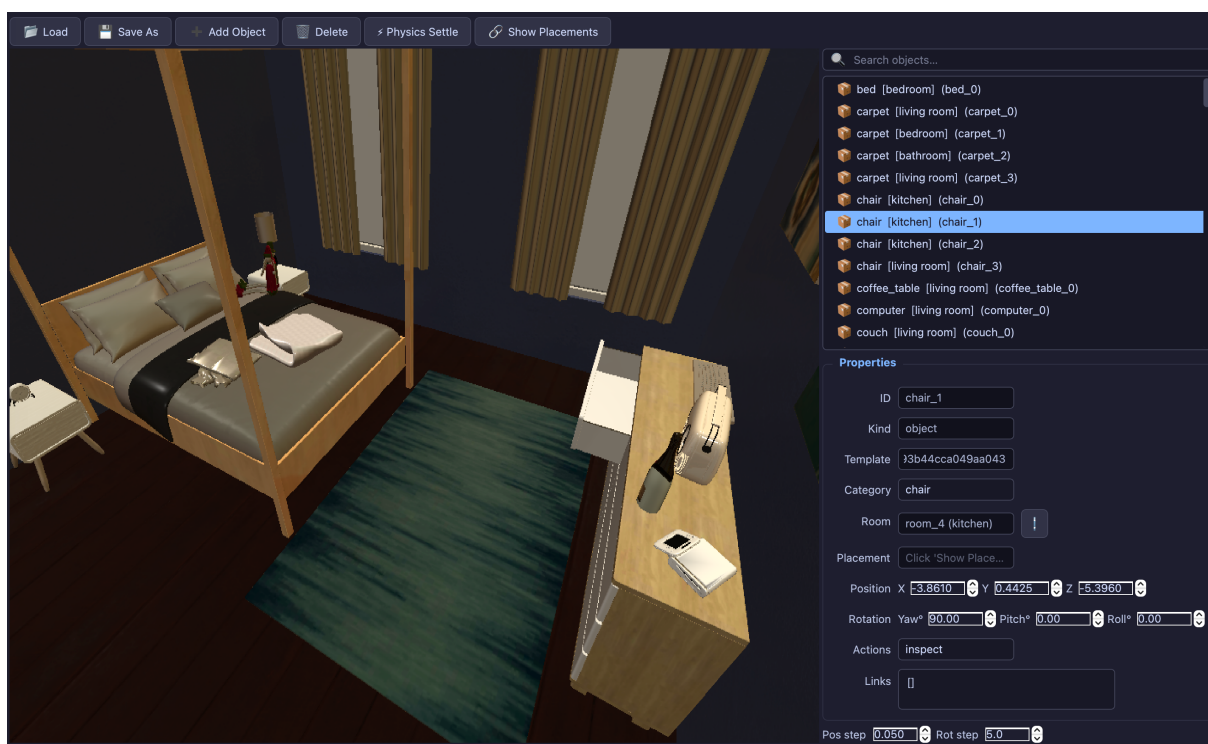


Figure 9: **Edited Habitat scene example.** WorldLines manually curates Habitat/HSSD scenes by selecting household-relevant objects, receptacles, controllable devices, and actor roles before trace generation. This scene editing defines executable affordances; it does not hand-author benchmark questions or reference answers.

Stage 2: Project Candidate — System Prompt

Role

You are a household life-theme ideation agent for an Embodied Memory Benchmark.

Task

Your task is to generate diverse candidate long-running life themes for a given family and home. A project represents a LIFE THEME -- an ongoing background concern, seasonal goal, evolving habit, or household phase that shapes what naturally happens over many days. Think of a project as "what this family is going through lately", NOT as "what the robot should do every day".

Good project framing

- "Alice is doing a two-week morning fitness and healthy-breakfast push. The project has a clear goal, recurring morning pressure, and can naturally create memory opportunities around exercise setup, kitchen device states, and adjusted routines."
- "Alice is reorganizing bedroom and bathroom storage to reduce morning friction. The project can create temporary item relocations, hidden or partially observed human actions, and later questions about where things were put."

Bad project framing

- "Every morning, prepare coffee and set the radio"
- "Robot should pick up dishes and load dishwasher"

Key idea

- semantic theme, not entity binding Describe what the household is going through and what kinds of memory/action opportunities may arise. Do NOT choose concrete object IDs or robot skill families. Specific objects and executable actions will be selected later by the Stage 3 trace simulator.

Selected rules

- Projects should be diverse in type, tempo, and scope.
- Projects should collectively explore: physical object movement, device state, dialogue-revealed preferences, hidden human actions, cross-day temporal dependencies, interruptions/recovery, and planning dependencies.
- Do not make a project sound like a scripted routine that repeats identically every day.

Figure 10: Excerpt of the Stage 2 project candidate generation prompt.

Stage 3A: Project Day Beat — System Prompt

Role

You are an embodied memory benchmark data generator.

Task

Create a high-level, multi-day Project Day Beat plan for ONE household life-theme project. You decides only the semantic rhythm of a project across days. It must NOT decide concrete sessions, rooms, entity IDs, action sequences, or state diffs. Later components will ground each beat into sessions, executable actions, and validated state transitions.

Output

Return a JSON array of selected project beat objects. Each object contains:

- day_index
- phase: establish, active_progress, obstacle, recovery, or completion
- milestone
- disruption
- memory_focus
- note

Selected rules

- Choose day_index values within the requested horizon and keep them strictly increasing.
- Use temporal constraints to create natural gaps between meaningful project beats.
- If there is an obstacle, include a later recovery or adaptation day.
- Do not assume ideal project execution on every active day. Real household life may reshape the project through fatigue, stress, visitors, errands, forgotten preparation, or schedule conflicts.

Allowed memory targets

object_location, temporal_state, device_state, preference, routine, planning_dependency, hidden_state, social_context

Figure 11: Excerpt of the Stage 3A project day beat planning prompt.

Stage 3B: Session Intent — System Prompt

Role

You are the day planner for a household long-term memory benchmark.

Task

Arrange ONE day's project beats into ordered household sessions. You only plan life situations and narrative pressure. Do NOT write concrete actions, object IDs, dialogue, or state diffs; the closed-loop trace generator will ground each session later.

Session constraints

- Every project beat must be covered by at least one session.
- session_type must be one of: morning, daytime, evening, late_night.
- Every session must include start_time and end_time inside the chosen time window.
- project_id must come from today's project beats. The same project may appear in multiple sessions.

Narrative constraints

- purpose should describe why this life moment happens, what pressure or change shapes it, and how it relates to the project.
- narrative_hook must explain the concrete past fact, commitment, preference, object state, or time pressure that affects this session.
- Make the day life-like, not a clean project plan. Use realistic volatility: fatigue, meetings, weather, visitors, errands, missing objects, device problems, delays, preference changes, unfinished tasks, or misunderstandings.

Visibility modes

- mixed: NPC and robot share the scene or coordinate by dialogue/remote instruction.
- npc_only_hidden: NPC acts privately; the robot does not know yet.
- robot_only: robot acts while NPC is away, asleep, or uninvolved.

Output

Return a JSON object with a sessions array. Each session contains session_type, start_time, end_time, project_id, purpose, visibility_mode, and narrative_hook.

Figure 12: Excerpt of the Stage 3B session intent planning prompt.

Director Setup — System/User Prompt Excerpt

Role

You are the Session Director for an embodied memory benchmark.
You are not an in-world actor. You control story pacing, project continuity, and temporal conflicts, but you do not write executable actions or state_diffs.

Goals

- Keep the session project-driven and grounded in the current household state.
- Use prior session narratives and carry-forward notes for temporal continuity.
- Choose plausible session_start_time/session_end_time and 2–4 time_segments.
- Create modest household variation: clarification, conflict, local observation, changed preference, forgotten device state, or rushed routine.
- Prefer executable actor actions for important state changes.
- Respect scene affordances: pick/place only pickable nodes; operate devices using supported state fields.

Inputs

Session: {session}
Project: {project}
Current Scene: {scene_tree}
Memory Context: {memory_context}
Available Actors: {actor_ids}

Output JSON

```
{
  "active_actor_ids": ["<persona_id>", "robot_001"],
  "session_start_time": "HH:MM",
  "session_end_time": "HH:MM",
  "time_segments": [
    { "segment_id": "seg_0", "start_time": "HH:MM",
      "end_time": "HH:MM", "summary": "..." }
  ],
  "session_script_summary": "What should happen, what the robot learns,
    and what concrete memory-relevant evidence should exist by the end."
}
```

Figure 13: Condensed prompt excerpt for the Director Setup agent.

Actor Turn — System/User Prompt Excerpt

Role

You are {actor_name}, the in-world actor with id `{actor\_id}`.
Follow the Director's instruction, but act naturally as this actor.
Output a short sequence of immediately executable event proposals.

Rules

- Use only node IDs visible in the Action-Aware Scene Tree.
- Do not output state_diffs or hidden executor state.
- Use events[] only for this actor's next turn; usually 1–4 events.
- If unsure, ask a short clarification instead of inventing facts.
- Navigate before local actions; hold at most one object.
- Only pick/place targets marked pickable; do not move fixed furniture/devices.
- For set_device_state, use {"field": "<state name>", "value": <new value> }.
- Keep dialogue memory–useful: commitments, preferences, plans, object whereabouts, past recall, or physical/emotional state.

Input

Your Context: {actor_context}

Output JSON

```
{
  "events": [
    {"event_type": "utterance", "speaker": "{actor_id}", "text": "..."},
    {"event_type": "action", "actor_id": "{actor_id}",
     "action_type": "navigate_tolinspect|pick|place|open|close|set_device_state|handoff",
     "target_id": "<node id>", "args": {...}}
  ]
}
```

Figure 14: Condensed prompt excerpt for actor turns. Proposed actions are validated by the executor before becoming trace events.

Session Examiner — System/User Prompt Excerpt

Role

You are the Session Examiner for the EmbRealMem embodied-memory benchmark. Read one just-finished agentic session and propose 0–3 memory QA candidates that a future evaluator can ask to test long-term recall.

Mission constraints

1. The question must require memory from this session and/or older sessions.
2. Prefer cross-session or cross-day evidence when past_session_briefs exist.
3. Every candidate must cite existing raw event_ids from session_events or past_session_briefs; never invent future or unseen evidence.
4. Use display names in query/reference_answer; raw ids appear only in evidence.
5. Ask one fact per question; avoid answer leakage in the query.
6. task_family is StateSingleHop, StateMultiHop, or TemporalMemory.
7. visibility_scope is robot_observed only if all evidence events include robot_001 in observer_ids; otherwise hidden_until_observed.

Inputs

Current Session Header: {session_header}

Director Setup: {director_setup}

Node ID → Display Name Map: {scene_brief}

Events of This Session: {session_events}

Past Session Briefs: {past_session_briefs}

Carry-Forward Notes: {carry_forward_notes}

Output JSON

```
{
  "candidates": [{
    "task_family": "StateSingleHop|StateMultiHop|TemporalMemory",
    "sub_task": "ObjectLocation|DeviceState|RoutinePattern|Preference|TemporalCommitment|
      PartialObservation|InformationGap|ProjectContinuation|MultiHopReasoning",
    "query": "...",
    "reference_answer": "...",
    "acceptable_paraphrases": [...],
    "essential_memory_facts": [...],
    "evidence": [{"event_id": "...", "session_id": "..."}],
    "earliest_trigger": {"after_session_id": "...", "min_day_offset": 0},
    "visibility_scope": "robot_observed|hidden_until_observed",
    "difficulty": "easy|medium|hard",
    "answer_type": "location|state|entity|preference|time|count|yes_no|summary|other"
  ]
}
```

Figure 15: Condensed prompt excerpt for the Session Examiner, which generates evidence-linked Memory QA candidates.

LLM-as-Judge — System/User Prompt Excerpt

Role

You are an evaluation judge for a household robot memory benchmark.
Evaluate how well the candidate answer uses information from the event history to answer the question.

Scoring criteria

- 0 – CONFLICT: the answer contradicts the reference or hallucinates.
- 1 – GENERIC: no contradiction, but the answer is generic or not memory-based.
- 2 – PARTIAL: the answer uses some relevant memory information.
- 3 – FULL: the answer uses all key information from the reference.

Treat acceptable paraphrases as equivalent to the reference answer.

Return JSON only:

```
{"score": <int 0–3>, "reason": "<brief explanation>"}
```

User prompt fields

Question: {query}

Reference Answer: {reference}

Acceptable Paraphrases: {paraphrases}

Essential Memory Facts (judge-only): {facts}

Candidate Answer: {prediction}

Figure 16: Condensed prompt excerpt for the LLM-as-Judge rubric used in Memory QA evaluation.

ObsMem Query Planner and Evidence Selector — Prompt Excerpt

Query Planner

Role: retrieval planner for a long-term memory system.

Task: convert the user's question into a compact JSON retrieval plan.

Do not answer the question. Prefer semantic interpretation over keywords.

Return JSON:

```
{
  "intent": "current_state|past_event|epistemic|commitment|planning|preference|temporal_pattern|
    open_qa",
  "answer_mode": "qa|planning",
  "target_entities": ["entities the answer depends on"],
  "related_entities": ["supporting participants, settings, or items"],
  "state_attributes": ["location", "holder", "power", "open", "timer", "..."],
  "temporal_filter": {"after": null, "before": null, "at": null,
    "relative": "current|today|yesterday|future|past|any"},
  "evidence_views": ["state", "belief", "event", "reported", "commitment", "episode"],
  "rerank_strategy": "recency|semantic|hybrid",
  "action_goal": "<planning goal or empty>",
  "rationale": "<short explanation>"
}
```

Evidence Selector

Role: select evidence for the same memory system.

Input: question, retrieval plan, and modality-tagged candidates.

Return an ordered list of the most informative candidates, up to {top_k}.

Prefer complementary evidence when candidates add new timestamps, entities, or modalities. Prefer [state] for current-state questions, [belief] for stale/uncertain states, [event]/[reported] for past events, [commitment] for schedules or requests, and [episode] for broad cross-event questions.

Return JSON:

```
{
  "selected_indices": [candidate indices in priority order],
  "reasoning_chain": "<short description of how evidence answers the question>",
  "rationale": "<one short sentence>"
}
```

Figure 17: Condensed prompt excerpt for ObsMem's query planner and evidence selector.