SubtleMemory: A Benchmark for Fine-Grained Relational Memory Discrimination in Long-Horizon AI Agents

Wenxuan Wang[1,2,*], Haoyu Sun[2,3,*], Fukuan Hou[4], Mingyang Song[5]
Weinan Zhang[1], Yu Cheng[2,7,†], Yang Yang[6,†]

[1]Harbin Institute of Technology, [2]Shanghai AI Laboratory
[3]Tongji University, [4]Xiamen University, [5]Fudan University
[6]Shanghai Jiao Tong University, [7]Nanyang Technological University

[*]Equal contribution. [†]Corresponding authors.

Abstract

Persistent AI assistants, such as OpenClaw, accumulate large collections of related memories over long-term interactions. As these memories grow, they may reinforce one another, diverge across contexts, or directly conflict, making correct assistance depend on memory relations rather than isolated recall. Existing long-term memory benchmarks do not systematically probe how agents preserve and utilize such relations during downstream tasks. To address this gap, we introduce **SubtleMemory**, a benchmark for fine-grained relational memory discrimination in long-running AI agents. SubtleMemory constructs relation-controlled latent semantic artifacts whose variants instantiate **complementary**, **nuanced**, or **contradictory** relations, and embeds them into realistic user-agent histories, requiring agents to recover distributed relational structures during later queries and instructions. The benchmark contains 1,522 evaluation instances over 10 long histories, grounded in 1,090 relation-controlled memory-variant sets and spanning user-related and non-user-related queries. Evaluating six standalone memory systems, two Claw-style agents with native memory modules, and three Claw-style agents with plugin memory modules, we find that current systems remain weak on fine-grained relational memory discrimination. We further introduce diagnostic protocols that reveal distinct capability profiles across memory preservation, retrieval, and downstream reasoning stages. Code and data are publicly available.

1 Introduction

Large language model agents are increasingly expected to function as persistent assistants rather than stateless conversational interfaces. In this setting, AI agents require memory mechanisms to retain and utilize information from long-term interactions to support continuity, personalization, and informed decision-making (Appendix A.1). As interactions accumulate over time, agents acquire large collections of highly related memories that may reinforce one another, subtly diverge under different contexts, or directly conflict. AI agents need to, therefore, not merely remember the past, but preserve and distinguish subtle relations among similar memories to behave appropriately.

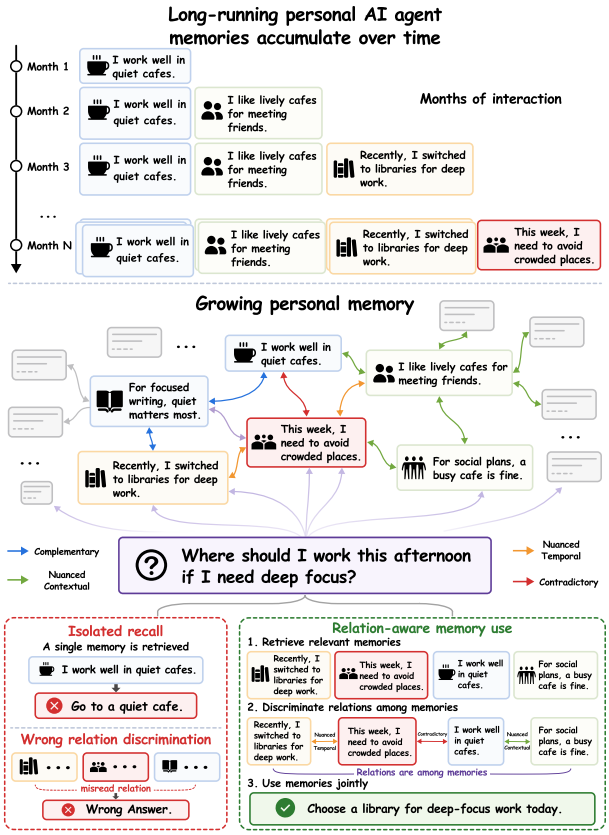

Figure 1: As personal memories accumulate, correct assistance depends on using relations among related memories rather than recalling isolated facts.

This challenge is not unique to AI agents. Human memory research has long observed that accumulated experiences can interfere with one another, particularly when memories are highly similar or context-dependent (Underwood, 1957; Johnson et al., 1993; Smith and Vela, 2001; Yassa and

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 25309–25359
October 24-29, 2026 ©2026 Association for Computational Linguistics

Stark, 2011; Schlichting and Preston, 2015; Hupbach et al., 2007). People may confuse where, when, or under what conditions a memory was formed, merge related experiences that should remain separate, or struggle to reconcile conflicting memories accumulated over time. Similarly, long-horizon AI agents may fail not only because of missing retrieval, but also similar or conflicting memories are incorrectly merged, overgeneralized, or misresolved despite being retrieved.

As discussed in Appendix A.2, earlier long-term memory benchmarks primarily evaluated the retrieval or manipulation of individual memories and generally did not explicitly control relations among multiple memories. More recent benchmarks, such as LongMemEval and ClawArena, partially or implicitly capture such interactions through multi-session reasoning, updating, or evolving histories, but do not systematically control them as an evaluation dimension. To make this dimension explicit and systematic, we introduce SubtleMemory, a comprehensive benchmark for fine-grained relational memory discrimination in long-term AI assistant usage, featuring explicit relation control and stage-specific diagnosis.

In SubtleMemory, we construct *relation-controlled semantic variants* associated with shared resolution targets. These variants instantiate **complementary, nuanced, or contradictory** relations that determine whether related memories should be aggregated, distinguished, or reconciled during downstream reasoning. Rather than exposing variants as explicit memory entries, we latently embed them into natural multi-turn user–agent interaction sessions distributed across long-horizon conversation histories. We then construct evaluation queries grounded in the underlying resolution targets and semantic variants, requiring agents to recover and correctly reason over semantically related memories scattered throughout the interaction history.

Beyond the benchmark itself, we contribute (1) a unified evaluation framework that supports standalone memory systems, framework-native memory agents, and plugin-based memory agents under a consistent evaluation protocol; (2) a task-level diagnostic framework that decomposes failures arising from memory construction, retrieval, and final response generation, enabling more precise analysis of where long-term memory systems break down; (3) through extensive evaluation across six standalone memory systems, two Claw-style agents with native memory modules, and three Claw-style agents with plugin memory modules, we show that current systems struggle both to preserve fine-grained relational information during memory formation and to retrieve sufficient task-relevant evidence during inference. More strikingly, our relation-wise evaluation and stage-wise diagnostics show that contradictory-memory instances expose more pronounced limitations across memory preservation, retrieval, and answer generation than complementary and nuanced instances, indicating that handling unresolved conflicts remains a key bottleneck for current long-term memory agents operating over long-horizon interaction histories.

2 Methodology

2.1 Preliminary Concepts and Relation Taxonomy

To characterize how multiple related memories should inform subsequent decisions, SubtleMemory emphasizes *relational memory reasoning*: the ability to aggregate compatible evidence, distinguish highly similar contexts, and reconcile conflicting memory records during downstream task completion. We first define the semantic primitives that underlie the benchmark construction process.

Semantic Seeds and Variants. We begin with a collection of semantic facts as the seeds to construct the benchmark. A **semantic seed** may describe either **user-related facts**, such as preferences, habits, identities, or plans; or **non-user facts**, such as world knowledge, object attributes, or contextual facts. An example of a user semantic seed is as follows:

> *Bonita prefers Japanese minimalist interior design.*

Given a semantic seed ϕ, we construct a set of **semantic variants** $V(\phi) = \{v_1, v_2, \ldots, v_n\}$, where each variant contextualizes or transforms the original seed through controlled operations such as detail enrichment, partial detail masking, or semantic neighboring search with closely related but non-identical content.

Resolution Target. To evaluate memory reasoning behavior, SubtleMemory introduces the notion of a **resolution target**, defined as an information need whose successful resolution requires reasoning over accumulated memories. For example, when a user asks an agent to generate an apartment renovation plan, the resolution target may require

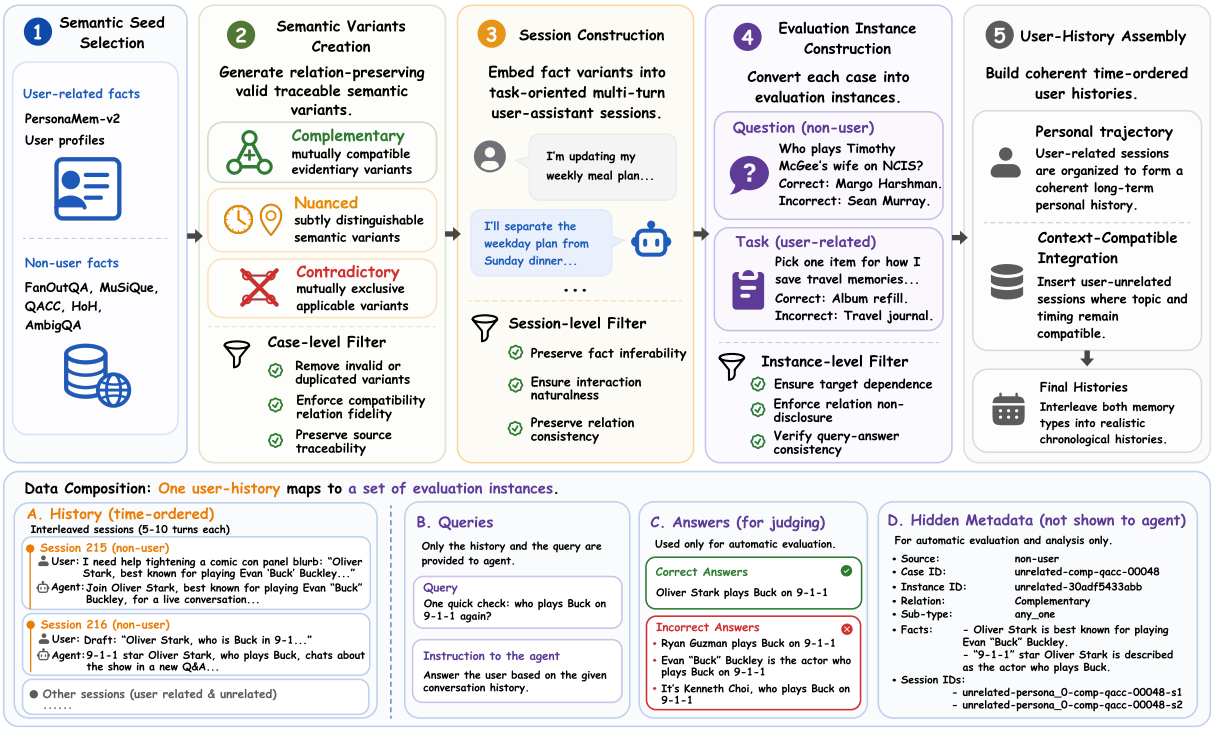

Figure 2: SubtleMemory builds each split through a five-stage pipeline that turns semantic seeds into relation-preserving variants, task-oriented sessions, evaluation instances, and time-ordered histories with judging metadata.

identifying the user's applicable interior-design preference from prior interactions.

Latent Semantic Artifacts. For each target τ, we determine its corresponding fact seed ϕ and select a subset of semantic variants $V_\tau \subseteq V(\phi)$, called the **target-conditioned semantic variant set**. These variants collectively participate in resolving the target after being instantiated into memories through interaction histories. We define compatibility relation types $r(V_\tau)$ for the items in the target-conditioned semantic variant set as:

- **Complementary**: variants provide mutually compatible evidence should be aggregated to resolve the target. During evaluation, the agent needs to either integrate information from multiple memory items or recognize any single evidence item is sufficient to answer the query. We further divide this relation into two subtypes: **Multi-evidence** and **Any-one**.
- **Nuanced**: variants are semantically similar but require fine-grained discrimination under the target. During evaluation, the agent must distinguish subtle differences between related memory items and identify the correct one for the query. Depending on whether the distinction arises from temporal or contextual cues, we further divide this relation into two subtypes: **Temporal** and **Contextual**.
- **Contradictory**: variants may appear applicable, but they cannot be jointly satisfied under the same target condition. During evaluation, the agent must recognize the underlying conflict and appropriately handle the resulting uncertainty from past experiences.

The resolution target τ, together with its semantic variant set V_τ and compatibility relation $r(V_\tau)$, forms a **latent semantic artifact** $(\tau,\ V_\tau,\ r(V_\tau))$ that is implicitly encoded into the benchmark and used to evaluate fine-grained memory discrimination in long-horizon interactions.

In Appendix B.1, we use an example target to illustrate the target-conditioned variants set under three types of relations.

2.2 Evaluation Overview and Taxonomy

User-history. We next operationalize semantic variants into long-horizon user-agent interaction histories. Given a resolution target τ and its target-conditioned variant set V_τ, each semantic variant is instantiated into a natural multi-turn user-agent conversation session. Rather than exposing facts directly as isolated memory entries, **the latent information gradually revealed itself through**

task-oriented conversations, such as planning discussions, preference clarifications, or contextual problem-solving interactions (Appendix B.2.1).

The instantiated sessions s_i are then distributed across a multi-session conversation history $\mathbf{H} = s_1, s_2, \ldots, s_m$, where semantically related variants are hidden in different sessions separated by unrelated sessions. This design simulates long-running personal AI assistant usage, in which relevant memories are naturally scattered across long interaction histories.

Memory Injection. For each memory system α, we replay a user history $\mathbf{H}$ by simulating its original memory formation process. Specifically, we first segment the interaction history into a sequence of history chunks $(H_1, H_2, \ldots, H_l)$ according to the memory formation granularity of the target system. For example, A-Mem (Xu et al., 2025) constructs memory units at the message level, while Mem0 (Chhikara et al., 2025) operates over message batches. We then sequentially feed these history chunks into the memory system following the original chronological order, allowing the system to incrementally construct its memory state as

$$M_t^\alpha = \mathcal{M}^\alpha(M_{t-1}^\alpha, H_t),$$

where $\mathcal{M}^\alpha$ and M_t^α denote the memory update function and memory state of system α, respectively. For simplicity, we use $M = M(\mathbf{H})$ to denote the final memory state after replaying the interaction history and omit α.

Evaluation Instance. Each evaluation instance is associated with a latent semantic artifact $(\tau, V_\tau, r(V_\tau))$ and a query q_τ that requires agents to recover and correctly reason over the hidden relational structure among semantically related memory items. When creating an evaluation instance, we use an LLM to generate the **reference correct answer set** A^+ and **wrong answer set** A^- by conditioning on $(\tau, V_\tau, r(V_\tau))$, and the subset of target-relevant sessions $\mathbf{H}_\tau$ from the user history:

$$(A^+, A^-) = \mathcal{G}_{\text{LLM}}\left(\tau,\, V_\tau,\, r(V_\tau),\, q_\tau, \mathbf{H}_\tau\right) \quad (1)$$

During evaluation, the agent first retrieves task-relevant memory evidence

$$m_\tau = \mathcal{R}(M, q_\tau),$$

where $\mathcal{R}$ denotes the memory retrieval procedure and $M(H)$ denotes the memory state constructed from the interaction history H. Using π to denote the response-generation model of the agent, the final answer is then generated as

$$a = \pi(q_\tau, m_\tau) = \pi(q_\tau, \mathcal{R}(M(\mathbf{H}), q_\tau)) \quad (2)$$

Comparing Eq.(2) to Eq.(1), we notice agents *never* observe the latent semantic artifacts directly, but only the entire, raw interaction history. Consequently, SubtleMemory evaluates whether agents can faithfully preserve fine-grained semantic details during memory construction, and subsequently retrieve, distinguish, and reconcile the relevant memories under long-horizon interactions.

Answer Correctness. We adopt an LLM-as-judge protocol to assign a binary correctness label based on whether the generated response correctly resolves the target implied by the query. The judge is provided with the latent semantic artifact $(\tau, V_\tau, r(V_\tau))$, the reference answer sets (A^+, A^-), and relation-specific evaluation guidelines.

2.3 Construction Pipeline

As shown in Figure 2, we construct the benchmark through a five-stage pipeline that progressively encodes latent semantic artifacts into realistic long-horizon user–agent interaction histories. Each stage includes dedicated verifiers and filters to maintain conversational naturalness while preserving the intended memory relations. Appendix C provides data construction details.

Stage 1: Semantic Seed Selection. We extract semantic fact seeds from high-quality open-source benchmarks. User-related seeds are derived from PersonaMem-v2 (Jiang et al., 2025) user preference profiles, while non-user seeds are drawn from knowledge-oriented QA benchmarks, including FanOutQA (Zhu et al., 2024), MuSiQue (Trivedi et al., 2022), QACC (Liu et al., 2025), HoH (Ouyang et al., 2025), and AmbigQA (Min et al., 2020).

Stage 2: Semantic Variants Creation. For user-related seeds, we first use an LLM to determine a candidate relation (complementary, nuanced, or contradictory) and the corresponding resolution target. Conditioned on the selected relation and target, the LLM then generates semantic variants through detail enrichment or selective detail omission. For non-user seeds, we conduct semantic neighborhood search over the seed corpus, identifying temporally

related, contextually specialized, or multi-hop dependent facts that naturally fit the target compatibility relation. Additional variants are generated through controlled detail omission.

Stage 3: Session Construction. For each fact variant, we generate a *task-oriented* multi-turn user–assistant interaction session in which the fact is revealed implicitly and progressively through conversation, better reflecting realistic assistant usage. To promote interaction diversity, we define 10 task categories, each paired with three workflow patterns reflecting different user-agent interaction styles.

Stage 4: Evaluation Instance Construction. For each latent semantic artifact, we generate evaluation queries together with reference correct and incorrect answers. Queries are designed to require relational reasoning over previously instantiated semantic variants. For non-user semantic facts, queries are formulated as knowledge-oriented questions. For user-related facts, queries are designed as either *structured form filling* or *resource arrangement* task, both of which naturally require agents to leverage user preference information accumulated through prior interactions. To verify this target dependence, we evaluate GPT-5.4 on all 815 non-user instances using only the query and obtain 1.23% accuracy, compared with 95.46% when we provide the annotated history evidence, indicating that these instances depend on interaction-history evidence (Appendix C.8).

Stage 5: User-history Assembly. Finally, we assemble sessions into long-horizon chronological user histories in which semantically related variants are distributed across separated interactions interleaved with unrelated sessions. This process produces realistic long-context histories where relevant evidence remains naturally scattered throughout the interaction stream.

2.4 Final Data Composition.

The final benchmark consists of 10 persona-level splits. Each split contains a long-horizon history of chronological user–agent interactions and a separate evaluation set grounded in latent semantic artifacts. In total, SubtleMemory contains 1,522 evaluation instances derived from 1,090 relation-controlled semantic variant sets, consisting of 361 complementary, 352 nuanced, and 377 contradictory sets. Each history contains an average of 236.4 memory-bearing sessions and 211.6K session tokens, creating long-context environments where target evidence is naturally interleaved with irrelevant or competing information. Queries span 10 domains, including culture, media, competition, world knowledge, society, cuisine, lifestyle, development, STEM, and nature. To assess benchmark-instance validity, we conduct a label-blind human audit of 150 stratified instances, yielding a 95.9% overall validation rate and 94.5% inter-annotator agreement (Appendix C.7).

3 Experiments

3.1 Experimental Setup

Evaluated Systems. We evaluate three deployment settings. (1) Six standalone memory systems: Mem0, MemOS, EverMemOS, MIRIX, A-Mem, and MemoBase (Chhikara et al., 2025; Li et al., 2025; Hu et al., 2026a; Wang and Chen, 2025; Xu et al., 2025; MemoBase, 2026). (2) Two Claw-style agents with native memory mechanisms: OpenClaw and MetaClaw (OpenClaw, 2026; Xia et al., 2026). (3) OpenClaw augmented with three plugin-based memory modules: Mem0, MemOS, and EverMemOS. Each system uses its default or author-recommended configuration, with detailed settings in Appendix D.1.1.

LLM-as-judge. We use Gemini 3.1 Pro Preview Thinking model as the LLM judge for answer evaluation. On a manually annotated benchmark containing 225 candidate answers, the judge achieves strong agreement with human annotations, with a Cohen's κ score of 0.963 (Appendix D.1.3).

Answer Generation. We tested two answer generation models: `gpt-5.4` (OpenAI, 2026) and `gpt-oss-120b` (OpenAI, 2025) in our main results. For all evaluated systems, we maintain consistency by using identical answer-generation instruction prompts (Appendix D.1.4). Standalone memory systems receive answer-generation instructions directly within the assembled context, whereas OpenClaw-based agents receive the same instructions through preloaded markdown instruction files.

Oracle Setting. For each evaluation query q_τ, we identify the user-agent sessions $\mathbf{H}_\tau$ in which the target semantic variants V_τ are latently encoded, and directly provide these sessions as retrieval results for answer generation. This setting bypasses the memory system's information extraction and retrieval processes, thereby approximating the upper-bound performance given perfect memory access.

Method	Complementary			Nuanced			Contradictory	All
	Multi-evidence	Any-one	Overall	Temporal	Contextual	Overall		
Base Model: GPT-5.4								
MemoBase	26.5%	43.3%	33.5%	33.3%	44.6%	38.9%	16.7%	32.1%
MIRIX	47.9%	52.0%	49.6%	51.8%	69.3%	60.5%	14.6%	46.5%
MemOS	51.2%	58.0%	54.0%	70.2%	77.8%	74.0%	23.9%	56.8%
Mem0	61.1%	61.3%	61.2%	81.3%	85.8%	83.5%	46.2%	69.0%
EverMemOS	63.5%	66.7%	64.8%	81.8%	87.4%	84.6%	37.1%	68.1%
A-Mem	58.3%	61.3%	59.6%	83.1%	85.3%	84.2%	**50.4%**	70.0%
MetaClaw	15.6%	20.0%	17.5%	17.7%	26.8%	22.2%	19.1%	20.3%
OpenClaw	62.6%	55.3%	59.6%	80.3%	83.0%	81.6%	25.5%	62.5%
MemOS + OpenClaw	57.3%	66.7%	61.2%	67.7%	80.2%	73.9%	15.9%	56.5%
Mem0 + OpenClaw	**70.1%**	61.3%	66.5%	**86.4%**	85.3%	85.8%	45.6%	**71.3%**
EverMemOS + OpenClaw	66.8%	**67.3%**	**67.0%**	85.4%	**89.7%**	**87.5%**	32.6%	69.1%
Oracle	79.6%	87.3%	82.8%	95.5%	93.8%	94.6%	68.7%	85.4%
Base Model: GPT-OSS-120B								
MemoBase	16.1%	38.7%	25.5%	27.8%	37.4%	32.5%	10.1%	25.3%
MIRIX	38.9%	47.3%	42.4%	42.7%	61.3%	51.9%	12.2%	39.8%
MemOS	48.8%	58.7%	52.9%	66.9%	69.8%	68.4%	18.0%	52.2%
Mem0	57.8%	58.7%	58.2%	80.1%	79.4%	79.7%	**40.6%**	**64.9%**
EverMemOS	53.1%	**60.7%**	56.2%	75.5%	81.7%	78.6%	26.8%	60.4%
A-Mem	58.3%	52.0%	55.7%	**80.3%**	80.4%	**80.4%**	37.7%	63.9%
MetaClaw	10.9%	16.7%	13.3%	13.6%	21.1%	17.3%	14.1%	15.6%
OpenClaw	46.4%	44.0%	45.4%	71.5%	69.8%	70.7%	20.7%	52.3%
MemOS + OpenClaw	40.8%	52.7%	45.7%	57.8%	70.6%	64.2%	15.9%	47.8%
Mem0 + OpenClaw	**60.2%**	57.3%	**59.0%**	72.2%	80.4%	76.3%	39.0%	62.9%
EverMemOS + OpenClaw	55.9%	57.3%	56.5%	74.2%	**82.7%**	78.4%	24.1%	59.8%
Oracle	82.0%	78.7%	80.6%	90.9%	88.7%	89.8%	41.6%	75.7%

Table 1: Main results on SubtleMemory. Answer correctness rates are grouped by answer-generation base model and reported by memory-relation type. Best/second non-oracle results per block bolded, underlined separately.

Perfect Retrieval Setting To disentangle retrieval quality from memory construction, we introduce a perfect retrieval setting. After memory system S builds its memory state from the complete user history, we provide the stored objects $\widetilde{m}_{\tau,S}$ written from $\mathbf{H}_\tau$ instead of raw sessions. This bypasses retrieval while preserving memory-construction effects. Appendix D.1.5 details the evidence source used in each setting.

3.2 Answer Generation Configuration

Under the oracle setting, answer generation performance depends primarily on the underlying model capability and prompting strategy. Using one complete user history from our benchmark (approximately 10% of the full benchmark), we evaluate three models: `gpt-4o-mini` (OpenAI, 2024), `gpt-oss-120b` (OpenAI, 2025), and `gpt-5.4` (OpenAI, 2026), under two prompting settings: a soft prompt with general guidance and a strong prompt with explicit instructions for target identification, conflict recognition, evidence fidelity, and clarification.

Table 2 shows that answer generation remains imperfect even under oracle memory access, suggesting that response generation itself constitutes a non-trivial limiting factor. Among all settings, `gpt-5.4` with the strong prompt achieves the best overall performance (90.1%), substantially outperforming both `gpt-4o-mini` and `gpt-oss-120b`, but contradictory-relation cases remain noticeably harder than complementary and nuanced cases.

Based on these results, we adopt `gpt-5.4` with the strong prompt as the default answer-generation configuration in our main experiments, while additionally reporting results using `gpt-oss-120b`. This choice controls answer-configuration variation across systems, allowing subsequent comparisons to reflect differences among memory mechanisms under a common answer-generation setting.

3.3 Main Results

Current memory systems remain substantially below Oracle performance on fine-grained relational memory discrimination. Across both base model settings, the strongest standalone sys-

Model	Prompt	Complementary			Nuanced			Contradictory	All
		Multi-evidence	Any-one	Overall	Temporal	Contextual	Overall		
GPT-4o-mini	Soft	**86.4%**	54.5%	75.8%	91.7%	94.4%	93.1%	8.3%	67.4%
	Strong	59.1%	54.5%	57.6%	69.4%	91.7%	80.6%	33.3%	63.1%
GPT-OSS-120B	Soft	77.3%	72.7%	75.8%	91.7%	97.2%	94.4%	11.1%	68.8%
	Strong	**86.4%**	63.6%	78.8%	94.4%	**100.0%**	97.2%	41.7%	78.7%
GPT-5.4	Soft	81.8%	**81.8%**	**81.8%**	**97.2%**	**100.0%**	**98.6%**	63.9%	85.8%
	Strong	81.8%	**81.8%**	**81.8%**	**97.2%**	**100.0%**	**98.6%**	**80.6%**	**90.1%**

Table 2: Oracle-setting answer-generation calibration using one complete user-history split containing 141 evaluation queries. Best values are bolded; second-best distinct values are underlined, with ties marked equally.

System	Setting	Comp.	Nuanced	Contr.	All
Base Model: GPT-5.4					
MemOS	Base	54.0	74.0	23.9	56.8
	+OpenClaw	61.2	73.9	15.9	56.5
	Δ	+7.2	-0.1	-8.0	-0.3
Mem0	Base	61.2	83.5	46.2	69.0
	+OpenClaw	66.5	85.8	45.6	71.3
	Δ	+5.3	+2.3	-0.6	+2.3
EverMemOS	Base	64.8	84.6	37.1	68.1
	+OpenClaw	67.0	87.5	32.6	69.1
	Δ	+2.2	+2.9	-4.5	+1.0
Base Model: GPT-OSS-120B					
MemOS	Base	52.9	68.4	18.0	52.2
	+OpenClaw	45.7	64.2	15.9	47.8
	Δ	-7.2	-4.2	-2.1	-4.4
Mem0	Base	58.2	79.7	40.6	64.9
	+OpenClaw	59.0	76.3	39.0	62.9
	Δ	+0.8	-3.4	-1.6	-2.0
EverMemOS	Base	56.2	78.6	26.8	60.4
	+OpenClaw	56.5	78.4	24.1	59.8
	Δ	+0.3	-0.2	-2.7	-0.6

Table 3: Effect of agent-runtime integration. Accuracy and Δ are reported in percentage points; darker blue/red cells indicate larger gains/losses.

tems are consistently Mem0, EverMemOS, and A-Mem. Under `gpt-5.4`, A-Mem achieves the best overall performance (70.0%), followed by Mem0 (69.0%) and EverMemOS (68.1%), yet still falls over 15 points behind the Oracle (85.4%). The gap remains substantial across all relation types, with the best standalone systems trailing Oracle by 18.0%, 10.0%, and 18.3% on complementary, nuanced, and contradictory relations, respectively. A similar trend holds under `gpt-oss-120b`.

Memory systems interact with agent context organization. Standalone memory systems generally outperform claw-style agents using their own native memory. Under `gpt-5.4`, native OpenClaw achieves only 62.5% overall performance, substantially below top standalone systems such as EverMemOS (68.1%) and Mem0 (69.0%). Integrating strong memory plugins substantially improves agent performance, raising OpenClaw to 69.1% with EverMemOS and 71.3% with Mem0. However, Table 3 further shows that the interaction between memory systems and agent context organization is not uniformly beneficial, but rather highly task- and model-dependent. Under `gpt-5.4`, agent-driven context organization improves complementary category for MemOS by 7.2%, but reduces contradictory category accuracy by 8.0% relative to the standalone system. Under the weaker `gpt-oss-120b` base model, adding the agent layer is generally harmful.

Contradictory relations constitute a greater challenge for current memory systems. Unlike nuanced relations, whose performance approaches saturation under Oracle evidence access, contradictory relations exhibit a larger system-to-Oracle gap under gpt-5.4, with GPT-5.4 achieving only 68.7% and `gpt-oss-120b` achieving 41.6%. Large gaps between tested systems and Oracle performance further suggest that current memory mechanisms face greater difficulty handling contradictory evidence. Under `gpt-5.4`, A-Mem achieves the strongest contradictory performance (50.4%), but still trails Oracle by 18.3 points.

Existing memory systems remain relatively weak at leveraging temporal information. Within nuanced relations, under `gpt-5.4` and `gpt-oss-120b`, 10 and 9 out of 11 evaluated systems, respectively, achieve higher performance on contextual-detail discrimination than on temporal discrimination, whereas Oracle exhibits the opposite pattern under both base models. This contrast suggests that temporal reasoning remains an important opportunity for improving temporal-aware memory organization.

4 Discussion

Because memory preservation and retrieval operate sequentially, end-task accuracy alone cannot localize failure sources. We therefore perform a staged waterfall analysis. Let S_O denote instances answered correctly under the oracle setting, filtering out answer-generation failures. Let $S_P \subseteq S_O$ denote instances that remain correct under perfect

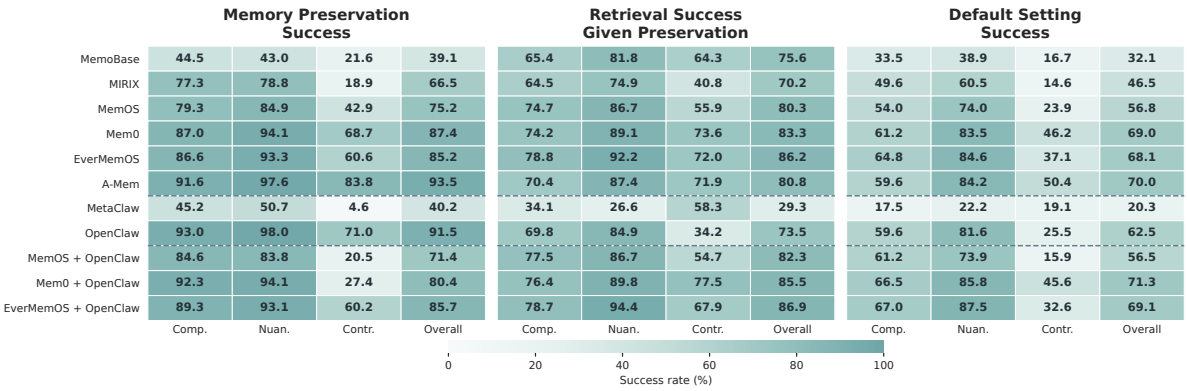

Figure 3: Diagnostic waterfall analysis of memory system performance. Overall performance is decomposed into stage-wise memory preservation success rate P_{preserve} and retrieval success rate conditioned on successful preservation P_{retrieve}. Rows correspond to the evaluated systems using GPT-5.4 as the answer generation model, sorted by overall accuracy.

retrieval setting, indicating sufficient information is preserved. Therefore, the memory preservation success rate can be defined as $P_{\text{preserve}} = \frac{|S_P|}{|S_Q|}$. Finally let $S_D \subseteq S_P$ denote instances that remain correct under the default setting. Since instances in S_P already contain sufficient preserved information, failures at this stage primarily reflect retrieval deficiencies. Hence the conditional retrieval success rate can be defined as $P_{\text{retrieve}} = \frac{|S_D|}{|S_P|}$.

Memory preservation and retrieval jointly determine downstream performance. As shown in Figure 3, both P_{preserve} and P_{retrieve} are generally correlated with final accuracy. For example, MemoBase achieves low preservation performance ($P_{\text{preserve}} = 39.1\%$), but relatively strong retrieval performance ($P_{\text{retrieve}} = 75.6\%$) , resulting in only 32.1% overall accuracy. The pattern becomes more evident when broken down by relation type. On contradictory-relation category, OpenClaw achieves strong memory preservation ($P_{\text{preserve}} = 71.0\%$), but weak retrieval performance ($P_{\text{retrieve}} = 34.2\%$), leading to only 25.5% final accuracy.

Raw interaction preservation could improve memory fidelity. A-Mem and OpenClaw are the strongest memory-preservation systems, achieving overall preservation rates of 93.5% and 91.5%, respectively. A unique shared characteristic is that, beyond maintaining structured memory states, both systems also preserve the original interaction sessions, which retain fine-grained cues that compressed memory abstractions may lose and help resolve SubtleMemory queries requiring detail preservation. In contrast, MetaClaw shows weak preservation performance overall (40.2%) and especially on contradictory cases (4.6%), together with low conditional retrieval performance (29.3%). Its memory mechanism emphasizes skill-like, experiential, and run-scoped abstractions, which are effective for reusable procedures but less aligned with factual relation-discrimination tasks that depend on exact details and competing records.

Empirical Observations and Insights. From Figure 3, we observe that different relations expose distinct memory bottlenecks. (1) Nuanced relations appear comparatively easier at the retrieval stage, potentially because they mainly require fine-grained discrimination among highly similar memories, where agents only need to identify the memory entry that best matches the target condition rather than retrieve all related evidence. (2) Complementary and Contradictory relations appear more retrieval-intensive, potentially because successful resolution often requires aggregating and reconciling multiple related memories. (3) Contradictory relations appear particularly challenging for memory preservation, potentially because mutually conflicting facts are more likely to interfere with each other inside the memory state.

5 Conclusion

We introduced SubtleMemory, a benchmark for evaluating fine-grained relational memory discrimination in long-horizon AI agents. To make this capability measurable, SubtleMemory implicitly embeds latent relation-controlled semantic artifacts into user histories, requiring agents to recover complementary, nuanced, and contradictory relations from natural interaction traces. Our results show that current systems do not simply fail by missing relevant memories, but also lose relation-critical

details, retrieve incomplete evidence, and struggle to use related memories correctly during answer generation. The diagnostic analysis further shows that these failures arise across answer generation, memory formation, and retrieval. We hope SubtleMemory provides a focused testbed for building memory systems that sustain subtle memory resolution throughout long-horizon interaction.

Limitations

Synthetic interaction histories. Despite our construction controls for increasing interaction diversity and reducing structural shortcuts, the inherent limitations of LLM generation mean that the resulting histories can only approximate naturally occurring interaction logs. Future work could incorporate naturally collected histories to better capture linguistic variability, behavioral irregularity, and evolving interaction patterns.

Controlled downstream interaction. To maintain a controllable answer space, SubtleMemory focuses on controlled single-output memory use and does not evaluate multi-turn downstream interaction or broader autonomous agent behavior. Future work could extend the benchmark to interactive task settings while preserving reliable evaluation.

Memory dynamics. Although evaluated systems process the fixed histories chronologically and dynamically update their memory states, the benchmark does not independently control repetition frequency, reinforcement strength, stability, or update rate. Future work could vary these factors independently to examine their effects on relation-sensitive memory use.

Evaluation scope. The current benchmark does not cover multimodal memories, multilingual interactions, or highly domain-specific workflows, and its results depend on the selected answer-generation and judging models. Future work could broaden these settings and evaluate a more diverse range of models and judges.

Ethical Considerations

Potential Risks. SubtleMemory is intended as an evaluation benchmark for long-term memory agents. As with other benchmarks, it may be over-optimized as a leaderboard target, which could limit generalization beyond the evaluated relation types and interaction settings. In addition, our automatic evaluation relies on LLM judges, which may introduce biases or occasional misjudgments.

Personally Identifying Information and Offensive Content. The benchmark is constructed from synthetic user-agent histories, LLM-generated annotations, and curated seed data from existing resources. Since seed data and generated interactions may contain sensitive, personally identifying, or inappropriate content, we apply filtering and manual inspection during construction to remove information that could name or uniquely identify individuals, as well as offensive or inappropriate content.

Instructions Given to Participants. This work did not involve recruited human participants, crowdworkers, or external annotators. Manual inspection and annotation were conducted by the authors, so no participant instructions, consent forms, or risk disclosures were required.

Acknowledgements

This work was supported by the Shanghai Artificial Intelligence Laboratory and Shanghai Jiao Tong University. This work was also supported by the National Science and Technology Major Project (Grant Nos. 2025ZD1606200 and 2025ZD1606203), the National Natural Science Foundation of China (Grant No. 92470205), and the Suzhou Science and Technology Development Program (SYG2024037).

References

Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2024. L-eval: Instituting standardized evaluation for long context language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14388–14411, Bangkok, Thailand. Association for Computational Linguistics.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

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. *Preprint*, arXiv:2601.06966.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready AI agents with scalable long-term memory. In *ECAI 2025*, Frontiers in Artificial Intelligence and Applications, pages 2993–3000. IOS Press.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. 2026. LightMem: Lightweight and efficient memory-augmented generation. In *The Fourteenth International Conference on Learning Representations*.

Zexue He, Yu Wang, Churan Zhi, Yuanzhe Hu, Tzu-Ping Chen, Lang Yin, Ze Chen, Tong Arthur Wu, Siru Ouyang, Zihan Wang, Jiaxin Pei, Julian McAuley, Yejin Choi, and Alex Pentland. 2026. MemoryArena: Benchmarking agent memory in interdependent multi-session agentic tasks. *Preprint*, arXiv:2602.16313.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*.

Chuanrui Hu, Xingze Gao, Zuyi Zhou, Dannong Xu, Yi Bai, Xintong Li, Hui Zhang, Tong Li, Chong Zhang, Lidong Bing, and Yafeng Deng. 2026a. EverMemOS: A self-organizing memory operating system for structured long-horizon reasoning. *Preprint*, arXiv:2601.02163.

Yuanzhe Hu, Yu Wang, and Julian McAuley. 2026b. Evaluating memory in LLM agents via incremental multi-turn interactions. In *The Fourteenth International Conference on Learning Representations*.

Almut Hupbach, Rebecca Gomez, Oliver Hardt, and Lynn Nadel. 2007. Reconsolidation of episodic memories: A subtle reminder triggers integration of new information. *Learning & Memory*, 14(1-2):47–53.

Haonian Ji, Kaiwen Xiong, Siwei Han, Peng Xia, Shi Qiu, Yiyang Zhou, Jiaqi Liu, Jinlong Li, Bingzhou Li, Zeyu Zheng, Cihang Xie, and Huaxiu Yao. 2026. ClawArena: Benchmarking AI agents in evolving information environments. *Preprint*, arXiv:2604.04202.

Bowen Jiang, Yuan Yuan, Maohao Shen, Zhuoqun Hao, Zhangchen Xu, Zichen Chen, Ziyi Liu, Anvesh Rao Vijjini, Jiashu He, Hanchao Yu, Radha Poovendran, Gregory Wornell, Lyle Ungar, Dan Roth, Sihao Chen, and Camillo Jose Taylor. 2025. PersonaMem-v2: Towards personalized intelligence via learning implicit user personas and agentic memory. *Preprint*, arXiv:2512.06688.

Marcia K. Johnson, Shahin Hashtroudi, and D. Stephen Lindsay. 1993. Source monitoring. *Psychological Bulletin*, 114(1):3–28.

Zhiyu Li, Shichao Song, Hanyu Wang, Simin Niu, Ding Chen, Jiawei Yang, Chenyang Xi, Huayi Lai, Jihao Zhao, Yezhaohui Wang, Junpeng Ren, Zehao Lin, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, Zhiqiang Yin, Qingchen Yu, Bo Tang, and 3 others. 2025. MemOS: An operating system for memory-augmented generation (MAG) in large language models. *Preprint*, arXiv:2505.22101.

Siyi Liu, Qiang Ning, Kishaloy Halder, Zheng Qi, Wei Xiao, Phu Mon Htut, Yi Zhang, Neha Anna John, Bonan Min, Yassine Benajiba, and Dan Roth. 2025. Open domain question answering with conflicting contexts. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1838–1854. Association for Computational Linguistics.

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, Bangkok, Thailand. Association for Computational Linguistics.

Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. 2023. CLIN: A continually learning language agent for rapid task adaptation and generalization. *Preprint*, arXiv:2310.10134.

MemoBase. 2026. MemoBase documentation. Accessed: 2026-05-18.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797. Association for Computational Linguistics.

OpenAI. 2024. GPT-4o mini: Advancing cost-efficient intelligence. Accessed: 2026-05-18.

OpenAI. 2025. gpt-oss-120b and gpt-oss-20b model card. Accessed: 2026-05-18.

OpenAI. 2026. Introducing GPT-5.4. Accessed: 2026-05-18.

OpenClaw. 2026. OpenClaw documentation. Accessed: 2026-05-18.

Jie Ouyang, Tingyue Pan, Mingyue Cheng, Ruiran Yan, Yucong Luo, Jiaying Lin, and Qi Liu. 2025. HoH: A dynamic benchmark for evaluating the impact of outdated information on retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6036–6063. Association for Computational Linguistics.

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.

Joon Sung Park, Joseph C. O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22. ACM.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025. Zep: A temporal knowledge graph architecture for agent memory. *Preprint*, arXiv:2501.13956.

Margaret L. Schlichting and Alison R. Preston. 2015. Memory integration: neural mechanisms and implications for behavior. *Current Opinion in Behavioral Sciences*, 1:1–8.

Ye Shen, Dun Pei, Yiqiu Guo, Junying Wang, Yijin Guo, Zicheng Zhang, Qi Jia, Jun Zhou, and Guangtao Zhai. 2026a. EvolMem: A cognitive-driven benchmark for multi-session dialogue memory. *Preprint*, arXiv:2601.03543.

Yiting Shen, Kun Li, Wei Zhou, and Songlin Hu. 2026b. Mem2ActBench: A benchmark for evaluating long-term memory utilization in task-oriented autonomous agents. *Preprint*, arXiv:2601.19935.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*.

Steven M. Smith and Edward Vela. 2001. Environmental context-dependent memory: A review and meta-analysis. *Psychonomic Bulletin & Review*, 8(2):203–220.

Haoran Sun, Shaoning Zeng, and Bob Zhang. 2026. H-MEM: Hierarchical memory for high-efficiency long-term reasoning in LLM agents. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 341–350, Rabat, Morocco. Association for Computational Linguistics.

Haoran Tan, Zeyu Zhang, Chen Ma, Xu Chen, Quanyu Dai, and Zhenhua Dong. 2025. MemBench: Towards more comprehensive evaluation on the memory of LLM-based agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19336–19352, Vienna, Austria. Association for Computational Linguistics.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Benton J. Underwood. 1957. Interference and forgetting. *Psychological Review*, 64(1):49–60.

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*.

Yu Wang and Xi Chen. 2025. MIRIX: Multi-agent memory system for LLM-based agents. *Preprint*, arXiv:2507.07957.

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*.

Peng Xia, Jianwen Chen, Xinyu Yang, Haoqin Tu, Jiaqi Liu, Kaiwen Xiong, Siwei Han, Shi Qiu, Haonian Ji, Yuyin Zhou, Zeyu Zheng, Cihang Xie, and Huaxiu Yao. 2026. MetaClaw: Just talk – an agent that meta-learns and evolves in the wild. *Preprint*, arXiv:2603.17187.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. A-Mem: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, volume 38, pages 17577–17604. Curran Associates, Inc.

Michael A. Yassa and Craig E.L. Stark. 2011. Pattern separation in the hippocampus. *Trends in Neurosciences*, 34(10):515–525.

Ningning Zhang, Xingxing Yang, Zhizhong Tan, Weiping Deng, and Wenyong Wang. 2026. HiMem: Hierarchical long-term memory for LLM long-horizon agents. *Preprint*, arXiv:2601.06377.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. ExpeL: LLM agents are experiential learners. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19632–19642.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. MemoryBank: Enhancing large language models with long-term memory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19724–19731.

Andrew Zhu, Alyssa Hwang, Liam Dugan, and Chris Callison-Burch. 2024. FanOutQA: A multi-hop, multi-document question answering benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 18–37. Association for Computational Linguistics.

Appendix Overview

This appendix provides supporting material for the benchmark definition, data construction process, and experimental analysis in the main paper.

- Appendix A situates SubtleMemory within prior work on memory-augmented LLM agents (Appendix A.1) and long-term memory benchmarks (Appendix A.2), then summarizes the benchmark-positioning comparison (Appendix A.3 and Table 4).
- Appendix B expands the task formulation, including the preliminary relation taxonomy (Appendix B.1), the evaluation overview and taxonomy (Appendix B.2), and examples showing how latent semantic variants are embedded into user-agent sessions (Appendix B.2.1).
- Appendix C documents the five-stage construction pipeline: semantic seed selection (Appendix C.1), semantic variant creation (Appendix C.2), session construction (Appendix C.3), evaluation-instance construction (Appendix C.4), and chronological user-history assembly (Appendix C.5). These sections also include the construction-model assignment, prompts, examples, and dataset statistics, together with human validation (Appendix C.7), target-dependence validation (Appendix C.8), and artifact availability, licenses, and intended use (Appendix C.9).
- Appendix D provides additional experimental details, including evaluated system settings (Appendix D.1.1), OpenClaw context organization (Appendix D.1.2), LLM-as-judge validation (Appendix D.1.3), answer-generation prompts (Appendix D.1.4), oracle and perfect-retrieval protocols (Appendix D.1.5), answer-model configuration (Appendix D.2), statistical analysis (Appendix D.3), repeated-run robustness (Appendix D.4), history-level reliability (Appendix D.5), answer-model and judge robustness (Appendix D.6), case studies (Appendix D.7), detailed perfect-retrieval results (Appendix D.8), and representative answer examples (Appendix D.9).
- Appendix E describes the roles of AI assistants in benchmark construction, evaluation, and manuscript editing.

A Related Work

A.1 Memory-augmented LLM agents

Long-term assistant agents use memory to keep prior interactions available as reusable agent state rather than transient context. Early agent systems explored external memory for personalization, reflection, experience reuse, and open-ended task continuation (Park et al., 2023; Zhong et al., 2024; Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2024; Majumder et al., 2023). More recent memory systems make memory management itself an explicit design problem. They study how memories should be written, consolidated, organized, updated, compressed, and retrieved through virtual-context memory managers, production memory services, hierarchical memory, temporal knowledge graphs, agentic note networks, multi-agent memory routing, memory-OS abstractions, and lightweight consolidation mechanisms (Packer et al., 2024; Chhikara et al., 2025; Rasmussen et al., 2025; Xu et al., 2025; Sun et al., 2026; Zhang et al., 2026; Wang and Chen, 2025; Li et al., 2025; Fang et al., 2026; Hu et al., 2026a). Claw-style agents introduce another deployment form, where memory can be maintained by the native agent runtime or supplied by plugin memory modules that inject recalled information into the runtime context (OpenClaw, 2026; Xia et al., 2026). These designs improve persistence, organization, and recall, but relation-sensitive behavior is usually left implicit in retrieval scores, summaries, links, or routing decisions. It therefore remains unclear whether current memory-augmented agents preserve and retrieve the distinctions among similar memories when those distinctions are required during downstream reasoning.

A.2 Benchmarks for long-term memory systems

Memory evaluation has moved from long-context use in static inputs to multi-session and agent-facing memory use. Long-context benchmarks measure retrieval and reasoning over long inputs (Bai et al., 2024; An et al., 2024; Hsieh et al., 2024). Long-term memory benchmarks then evaluate whether agents can retain conversation history, update user preferences, apply personalized facts, perform memory operations, or use stored state during later tasks (Maharana et al., 2024; Wu et al., 2025; Jiang et al., 2025; Tan et al., 2025; Hu et al., 2026b; Bian et al., 2026). More recent benchmarks further stress dynamic memory use in multi-session, task-oriented, and evolving information environments (Shen et al., 2026a,b; He et al., 2026; Ji et al., 2026), including ClawArena's emphasis on multi-source conflict, belief revision,

and implicit personalization over long-horizon interactions. Together, these benchmarks broaden long-term memory evaluation, with some partially or implicitly capturing interactions among multiple memories through multi-session reasoning, updating, or evolving histories. However, they do not systematically define and control how multiple memories relevant to the same downstream target should be aggregated, distinguished, or retained as unresolved conflict.

A.3 Positioning of SubtleMemory

To fill this gap, we introduce SubtleMemory, a benchmark for evaluating fine-grained relational memory discrimination during downstream reasoning. SubtleMemory organizes each instance around a resolution target and a target-conditioned semantic variant set, and explicitly controls whether the relevant memories are complementary, nuanced, or contradictory. Table 4 compares SubtleMemory with representative benchmarks across different styles of long-term memory evaluation. The comparison dimensions are:

- *Memory source*, *interaction form*, and *primary evaluation target* describe the benchmark input, interaction setting, and task.
- *Controlled dependency* names the dependency explicitly constructed to determine the answer.
- The relation rows ask whether the benchmark explicitly tests target-conditioned related memories, complementary aggregation, nuanced context/-time discrimination, and contradictory conflict preservation.
- The diagnostic rows compare whether the benchmark separates preservation, retrieval, and answer-generation behavior.

B Methodology

B.1 Preliminary Concepts and Relation Taxonomy

Consider the semantic seed:

> *Bonita prefers Japanese minimalist interior design.*

For the resolution target of identifying Bonita's applicable interior-design preference for an apartment renovation plan, SubtleMemory can construct different target-conditioned semantic variant sets around the same seed depending on the intended memory relation.

Complementary relation. The variant set may include:

- *Bonita prefers Japanese minimalist interiors with light wood furniture.*
- *Bonita prefers minimalist interiors with neutral color palettes.*
- *Bonita prefers low-clutter rooms with concealed storage.*

These memories are mutually compatible, so the agent should aggregate them into a single renovation brief.

Nuanced relation. The variant set may include:

- *Bonita prefers Japanese minimalist interiors for her apartment.*
- *Bonita prefers Scandinavian minimalist layouts for workshop spaces.*
- *Bonita prefers bold industrial styling for pop-up exhibition booths.*

These memories are topically similar, but the apartment-renovation target selects only the home-interior preference.

Contradictory relation. The variant set may include:

- *Bonita prefers Japanese minimalist interiors for the apartment renovation.*
- *Bonita no longer wants Japanese minimalist interiors for the apartment and now prefers a maximalist vintage style.*

These memories cannot be jointly satisfied under the same target, so the agent should recognize the conflict and ask for clarification instead of silently choosing one side.

B.2 Evaluation Overview and Taxonomy

B.2.1 Embedding Semantic Variants into User Sessions

Semantic variants are not exposed to the evaluated agent as isolated memory entries. Instead, each variant is embedded into a natural task-oriented user-agent session, where the relevant information becomes recoverable from the user's goal, constraints, corrections, and concrete details. The following two examples show this process for a user-related nuanced-relation case and a non-user

Comparison dimension	LoCoMo	LongMemEval	PersonaMem-v2	Mem2ActBench	ClawArena	SubtleMemory
Memory source	User dialog	User history	Persona prefs.	Tool history	Workspace evidence	**User + external**
Interaction form	Multi-session	Sessions + distractors	Long personalization	Interrupted tools	Evolving rounds	**Task sessions**
Primary evaluation target	Recall/summary	QA/abstention	Preference inference	Tool action	Conflict/revision	**Relation-sensitive answer**
Controlled dependency	Implicit evidence	Evidence/update	Preference evolution	Memory chain	Source/update/pref.	**Target-conditioned variants**
Target-conditioned related memories	◗	◗	✗	◗	◗	✓
Complementary aggregation	◗	◗	✗	◗	◗	✓
Nuanced context/time discrimination	◗	◗	◗	✗	◗	✓
Contradictory conflict preservation	✗	◗	✗	◗	✓	✓
Explicit relation taxonomy	✗	✗	✗	✗	◗	✓
Preservation/retrieval/answer diagnostics	✗	◗	✗	◗	◗	✓
Main diagnostic granularity	Answer	Retrieval+answer	Preference type	Tool+retrieval	Category reliability	**Stage waterfall**

Table 4: Transposed feature comparison between SubtleMemory and representative long-term memory benchmarks from different evaluation styles: multi-session conversation (LoCoMo (Maharana et al., 2024)), long-term memory QA (LongMemEval (Wu et al., 2025)), personalization (PersonaMem-v2 (Jiang et al., 2025)), memory-grounded tool use (Mem2ActBench (Shen et al., 2026b)), and Claw-style agent evaluation (ClawArena (Ji et al., 2026)). ✓ denotes an explicit benchmark feature, ◗ denotes partial or implicit coverage, and ✗ denotes that the feature is not a primary target.

complementary-relation case. Figure 4 shows the user-related case, and Figure 5 shows the non-user case.

C Data Construction

This section records the concrete source criteria, prompt interfaces, filtering rules, representative examples, and final composition used to instantiate the construction pipeline.

Construction model assignment. Stage 1 selects semantic seeds from source datasets, while Stage 5 assembles the accepted sessions into chronological histories; neither stage introduces new LLM-generated content. In Stages 2–4, we use `gpt-5.4` for relation planning and all generation steps, including semantic-variant generation, session construction, and evaluation-query and answer-candidate generation. We use `gemini-3.1-pro-preview-thinking` for all LLM-based filters applied to semantic variants, sessions, queries, and answer candidates. Both models use temperature 0, and `gpt-5.4` uses a maximum output length of 16,384 tokens. We assign generation and filtering to different model families to reduce same-family self-validation. Table 12 reports the model identifiers and settings, while the stage-specific prompts are provided in the following subsections.

C.1 Stage 1: Semantic Seed Selection

User-related seed selection. User-related seeds come from PersonaMem-v2 (Jiang et al., 2025). We retain the following source fields and selection metadata:

- We select ten personas with distinct backgrounds and application domains (Table 5).
- The retained seed unit is a persona, topic, and source preference.
- We keep the sanitized profile as user background, remove raw conversations and update metadata, and use the active preference as the semantic seed.
- Figure 6 shows one retained profile and its preference fields.

Non-user seed selection. Non-user seeds are retained only when their evidence structure can support relation-controlled variants. The retained source filters are:

- FanOutQA (Zhu et al., 2024): development samples with at least five linked evidence or subquestion items.
- MuSiQue (Trivedi et al., 2022): answer-classified development samples with four linked evidence items, after duplicate-heavy

User-related Task-oriented Session Embedding

Case. Amara uses different design styles depending on whether she is shaping her home environment or creating materials and spaces for kids' science activities.
Target-conditioned variants.

- At home, Amara prefers minimalist, Scandinavian-inspired design with clean lines, light wood tones, and an uncluttered feel.
- When designing STEM workshops or toy car race setups for children, Amara intentionally uses bold colors, interactive visual cues, and playful layouts to keep the space energetic and engaging.

Session embedding. Each variant is assigned to a separate task-oriented session. The session does not state the variant as a direct memory sentence; instead, it makes the variant recoverable through the user's task and interaction details.

Session 1: `workflow_setup`

Assigned variant. Home-design preference.

User. I need a repeatable way to stop overbuying decor for my apartment. Cute objects keep sneaking in and then the room starts feeling visually noisy.

Assistant. A good fix is a tiny system instead of relying on willpower. . . .

User. The things that keep surviving every cleanup are the pieces with simple lines, lighter wood, soft neutrals, maybe one muted green. When tabletops get crowded, my brain does too.

Assistant. That gives us a very clear filter. Here's a first pass. . . .

Session 2: `information_organization`

Assigned variant. Workshop-design preference.

User. My notes for a kids' toy-car physics event look like a lab notebook went through turbulence. I need help turning the pile into a clean structure.

Assistant. Let's turn the turbulence into lanes. A good first sort would be: . . .

User. Yes, that split is closer. A few non-negotiables: bright lane colors, big picture-based instructions, arrows or footprints that tell the cars-and-kids where to move, and separate little track islands instead of one long teacher table.

Assistant. Absolutely. Based on what you've said, here's the cleaner sort: . . .

Figure 4: User-related session embedding for a nuanced contextual variant set. Both sessions are topically about design, but only the home-design session should guide an apartment-design request.

Persona ID	Selected Persona	Domain Coverage
0	Amara Nwosu	STEM education, physics, mechanical engineering, youth outreach
334	Dr. Suthida Rattanaporn	Healthcare, nursing education, curriculum design
351	Gareth McAllister	Agriculture, rural work, family business, stress management
341	Malik Thompson	Art, politics, social justice, urban culture
92	Julian Marcus Ortega	NLP, computational social science, social media
637	Lars Henriksen	Professional sports, endurance training, performance optimization
45	Isabel Moraes Santiago	Elderly persona, cultural heritage, language preservation
996	Christopher "Chris" Halberg	Law, institutional accountability, professional fallibility
135	Adriana Ionescu	Migration, cross-cultural adaptation, urban planning
144	Dr. Amira Suryani	Neuroscience, speculative technology, ethics, consciousness research

Table 5: Selected PersonaMem-v2 personas used for user-related semantic seed selection. The set increases coverage across demographic backgrounds, occupations, cultures, and application domains.

items are cleaned down to 153 retained samples.

- QACC (Liu et al., 2025): 796 retained records from roughly 1.7K candidates, requiring multiple entries with the same answer surface form.
- AmbigQA (Min et al., 2020): records with at least four QA pairs for context-conditioned candidates after manual filtering, and records with exactly three QA pairs for contradiction candidates.
- HoH (Ouyang et al., 2025): records with at least three outdated information entries for temporal variants.

Figure 7 shows one retained non-user source record and the selected semantic facts.

C.2 Stage 2: Semantic Variant Creation

User-related semantic variant creation. For user-related variants, the planner first assigns one compatibility relation type and subtype to each source preference under target count constraints. The generator then receives the planned relation,

Non-user Task-oriented Session Embedding

Case. The answer requires matching several Korean metropolitan cities with their symbol flowers.
Target-conditioned variants.

- Busan's symbol flower is Camellia.
- Ulsan's symbol flower is Pear flower.
- Incheon's symbol flower is Rose.

Session embedding. Each variant is assigned to a task-oriented session. The session does not state the variants as a benchmark list; instead, it makes the target facts recoverable through ordinary learning and artifact-production tasks.

Session 1: `learning_explanation`

Assigned variant. Busan and Ulsan symbol-flower facts.

User. I'm helping my younger cousin study Korean city symbols. Can you explain in simple terms what a city's symbol flower is, and how to make the idea easy to remember?

Assistant. A city's symbol flower is an official flower the city uses to represent itself, a bit like a visual emblem. It usually reflects the image the city wants to convey, so the easiest way to remember it is: not "favorite flower," but "flower that stands for the city."

User. That helps, but I learn better with examples. Could you use one metropolitan city so I can hear how you'd explain it out loud?

Assistant. Sure. If I were explaining it with Busan, I'd say that Busan's symbol flower is Camellia. Then I'd add that this is just one example of how a city chooses a flower to act as part of its civic identity, so you're connecting a place name with a representative image.

User. Nice. Could you give me a second city too? I think comparing two examples would make the concept stick better.

Assistant. Yes, comparison works well here. You could pair Busan with Ulsan and point out that Ulsan's symbol flower is Pear flower. That shows the main idea: each city has its own emblematic flower, so the point isn't to memorize flowers in general, but to see that the symbol belongs to that specific city.

User. Good approach. One last thing: can you give me a memory trick for the Ulsan example? That's the one I'm more likely to forget.

Assistant. A simple trick is to link the word "pear" with the image of a light spring blossom. If you picture Ulsan paired with a pear blossom image, it becomes easier to recall that Ulsan's symbol flower is Pear flower.

Session 2: `artifact_production`

Assigned variant. Incheon symbol-flower fact.

User. I'm putting together a small cultural display and need a short caption for an Incheon profile card. Can you help me draft one?

Assistant. Sure — here's a clean starting option: "Incheon, a dynamic gateway city, blends modern energy with local pride and tradition."

User. Nice start. I want it to feel a little more specific though, like something that belongs on a city handout card.

Assistant. Absolutely. Here's a more specific version: "Incheon is a vibrant city known for its forward-looking spirit; its symbol flower is the rose, reflecting beauty and civic pride."

...

User. I like that. Could you give me two final options: one plain and one a little more polished? Keep both short.

Assistant. Sure: Plain: "Incheon's symbol flower is the rose." Polished: "The rose, Incheon's symbol flower, reflects the city's beauty and pride."

User. Great, I'll use the polished one. Can you give it to me one last time with no extra commentary so I can paste it into the layout?

Assistant. The rose, Incheon's symbol flower, reflects the city's beauty and pride.

Figure 5: Non-user session embedding for a complementary multi-evidence variant set. The required facts are distributed across ordinary task-oriented interactions rather than exposed as a benchmark list.

User-related source record and raw preferences

Selected example persona profile fields.

`persona_id`: 0; `name`: Amara Nwosu.

`short_persona`: A science whiz who loves to incorporate physics concepts into unique toy car races.

`age`: 28; `gender`: Female; `nationality`: Nigerian-American; `languages_spoken`: English, Igbo, French.

`personality.traits`: curious, playfully competitive, innovative, patient, analytical, warm.

`personality.hobbies`: custom racetracks, aerodynamics in miniature models, science fairs, vintage toy cars, astronomy stargazing nights.

`personality.quirks`: names toy cars after famous scientists, keeps a small chalkboard for equations, and uses physics analogies casually.

`occupation`: STEM Education Program Designer at the Smithsonian Science Education Center, focused on interactive STEM workshops, hands-on physics activities, and museum or school outreach.

Additional fields include `education`, `demographics`, `speaking_style`, `therapy_background`, `health_and_medical_conditions`, and `sensitive_information`.

Raw PersonaMem-v2 preference fields.

`stereotypical_preferences`: "Values strong ties to extended family traditions," "Loves spicy jollof rice and suya," "Follows prominent African diaspora authors and speakers," "Engages in traditional Nigerian holiday celebrations," "Reads African literature classics," "Participates in cultural dance workshops," "Enjoys drumming rhythms in community events," ...

`anti_stereotypical_preferences`: "Enjoys minimalist and Scandinavian-inspired home design," "Prefers quiet coffee shops over loud social events," "Loves indie folk and ambient music," "Enjoys watching Nordic noir detective series," "Prefers muted neutral fabrics in clothing," "Practices yoga in tranquil environments," ...

`neutral_preferences`: "Enjoys reading science fiction novels," "Loves homemade pizza nights," "Appreciates well-written mystery novels," "Likes learning new languages casually," "Prefers reusable eco-friendly shopping bags," "Likes solving crosswords and logic puzzles," "Enjoys planning and organizing events," ...

Figure 6: Profile fields and raw PersonaMem-v2 preference fields used as user-related semantic seeds for the selected example persona. Additional fields and preferences are omitted for space.

Non-user source record and selected facts

Raw source shape. External-knowledge inputs contain a source-style question and evidence items. One retained source question asks, "What are the names of six Metropolitan cities in Korea and their respective symbol flowers?"
Selected semantic facts after fact selection. Busan's symbol flower is Camellia. Incheon's symbol flower is Rose. Ulsan's symbol flower is Pear flower.

Figure 7: Non-user source question and selected external-knowledge facts used as semantic seeds for variant creation.

source preference, persona, relation definitions, and boundary rules, and must return the same relation with a concise case description and one-sentence variants. The filter removes unsupported, duplicated, mislabeled, internally invalid, unnatural, or order-dependent variant sets before any session is written. Figures 8, 9, and 10 show the prompts, and Figure 11 shows accepted examples.

Non-user semantic variant creation. For non-user variants, relation-specific fact selection follows the source family:

- FanOutQA and MuSiQue records become *Multi-evidence* sets whose target requires combining selected facts.
- QACC records become *Any-one* sets because repeated surface forms independently support the same answer.
- AmbigQA context records become *Contextual* sets, and HoH dated records become *Temporal* sets.
- AmbigQA three-answer records become contradictory-relation sets by selecting two conflicting QA entries and removing qualifiers that would reconcile them.

Figures 13, 14, and 15 show the relation-specific fact-selection prompts, and Figure 12 shows accepted non-user variant sets.

C.3 Stage 3: Session Construction

User-related session construction. Each generated user-related session may express only its assigned variant, while other same-set variants are provided only as relation context. The assigned variant is hidden supervision rather than a memory sentence, so the conversation must make the target recoverable through concrete signals such as polarity, behavior, boundary, example, routine, correction, artifact detail, or use context. The opening user turn cannot carry all target evidence, and task diversity is sampled over conversation type, workflow rhythm, message count, persona-signal level, and same-set session history (Table 6). The session filter rejects unnatural, mechanical, hidden, over-explicit, type-mismatched, label-revealing, or relation-breaking sessions. Figures 16 and 17 show the prompts, and Figure 18 shows an accepted session set.

Non-user session construction. A planning step assigns selected external facts to independent sessions, chooses a conversation type and workflow rhythm for each session, and keeps facts assigned to other sessions unmentioned. Persona-signal levels are not sampled, because these facts are external knowledge rather than persona-specific preferences or states. *Multi-evidence* cases keep partial evidence separate until query time, *Any-one* cases keep equivalent facts redundant but natural, *Contextual* and *Temporal* cases preserve the decisive qualifier, and contradictory-relation cases embed one target claim per session without announcing or resolving the conflict. Figure 19 shows the planning and generation prompts, and Figure 20 shows real non-user session excerpts.

C.4 Stage 4: Evaluation Instance Construction

User-related evaluation instance construction. User-related queries use either `structured_form`, where fixed fields make values and clarification behavior judgeable, or `resource_arrangement`, where candidate resources are provided and the agent selects, ranks, excludes, assigns, or finalizes them. Each q_τ must be self-contained, realistic, and underdetermined without the target-conditioned variants. Any time, context, role, location, scope, or object condition must be neutral rather than answer-revealing. Correct answers A^+ encode required fact alignment, condition, candidate selection, field value, exclusion, or clarification, while incorrect answers A^- instantiate relation-specific failures. The instance filter removes broad, leaky, unsupported, relation-inconsistent, or unclearly judgeable instances. Figures 21, 22, and 23 show the prompts, and Figure 24 shows both task forms.

Non-user evaluation instance construction. Non-user evaluation instances add relation-specific query and answer rules:

- *Multi-evidence* questions preserve the hidden dependency structure without enumerating every hop.
- *Contextual* questions specify exactly one decisive context, while *Temporal* questions cover explicit, event-anchored, and relative temporal conditions.
- Contradictory-relation questions target the disputed point without adding qualifiers that would resolve the conflict.

Conversation type	Primary agent task	Sampled workflow options
decision_support	Compare options, weigh tradeoffs, clarify criteria, and decide.	*1.* Start with a practical choice, then let criteria, tradeoffs, and a final boundary emerge. *2.* Move from a rough dilemma to narrowed options, adding one constraint at a time. *3.* Let the assistant compare options, the user correct weighting, and the ending settle on a next action.
planning_coordination	Organize steps, timing, dependencies, preparation, responsibilities, or contingencies.	*1.* Start with an upcoming task, then build sequence, dependencies, and timing. *2.* Let the assistant sketch a plan, the user reveal constraints, and the final turn tighten it. *3.* Move from scattered preparation to an order of operations with responsibilities or contingencies clarified late.
troubleshooting	Diagnose a practical problem and work toward next steps.	*1.* Start with a symptom or blocked task, then narrow likely causes through observations. *2.* Let the assistant propose a check, the user report what is true, and the session end with a fix path. *3.* Move from confusion to diagnosis, with each user turn adding one clue, failed attempt, or condition.
learning_explanation	Explain a concept, mechanism, background, example, or applied understanding.	*1.* Start with a concept or situation, then refine the explanation through examples and limits. *2.* Let the assistant explain simply, the user ask for an applied version, and the final answer connect it to use. *3.* Move from background explanation to a concrete example, with the user specifying what kind of understanding helps.
resource_selection	Choose among tools, places, courses, materials, activities, routes, services, or other resources.	*1.* Start with a need for options, then reveal constraints, criteria, and reasons to rule choices in or out. *2.* Let the assistant suggest categories, the user narrow the context, and the ending recommend a short ranked set. *3.* Move from broad resource search to best fit by adding budget, setting, time, skill, access, or taste constraints.
workflow_setup	Build a repeatable routine, checklist, tracking approach, template, or operating pattern.	*1.* Start with a recurring friction point, then turn it into a routine, checklist, or tracking pattern. *2.* Let the assistant propose a structure, the user identify what would break it, and the final answer simplify it. *3.* Move from a messy process to a usable template, with each user turn adding one real-world constraint.
information_organization	Sort notes, options, evidence, constraints, ideas, or action items into a clearer structure.	*1.* Start with scattered notes, options, or evidence, then sort them into categories, priorities, or action items. *2.* Let the assistant impose an initial structure, the user add missing context, and the final answer reorganize it. *3.* Move from an unstructured list to a clearer map, with the user clarifying what distinction matters most.
personal_reflection	Reason about personal patterns, boundaries, preferences, energy, values, or decision principles.	*1.* Start with a noticed pattern, then use concrete examples to name a boundary or decision principle. *2.* Let the assistant offer an interpretation, the user correct part of it, and the session land on a usable framing. *3.* Move from a vague feeling to a grounded self-read, without turning the session into generic advice.
artifact_production	Produce a new text or content artifact, such as an email, message, post, paragraph, or script.	*1.* Start with a simple artifact request, then reveal tone, audience, and constraints before the final text. *2.* Let the assistant draft first, the user add practical constraints, and the final version become more specific. *3.* Move from purpose to usable artifact, with revisions focused rather than repeatedly reworking the same text.
artifact_review_or_localization	Revise, shorten, translate, localize, or change the tone of an existing artifact.	*1.* Start from an existing artifact, then revise purpose, audience, tone, or language through targeted changes. *2.* Let the assistant diagnose what is not working, the user clarify the desired effect, and the final version adjust. *3.* Move from review to localized or tightened output, with the user adding context gradually.

Table 6: Conversation types and sampled workflow options used for session diversity. Each session samples four candidate conversation types, and each candidate type is paired with one sampled loose workflow rhythm rather than a fixed template.

User-related relation-planning prompt excerpt

"Plan the relation type and subtype for each source preference before semantic variant generation." "Persona: ..." "Persona id: ..." "Relation type definitions: ..." "Relation subtype definitions: ..." "Boundary rules: ..." "Target relation type counts: ..." "Subtype balancing policy: ..." "Preferences: ..."
"Requirements:"

- Assign exactly one `relation_type` and `relation_subtype` to every `preference_id`.
- Match the target relation type counts exactly.
- Within each `relation_type`, split `relation_subtype` assignments as evenly as possible following the fixed subtype order above.
- Use the fixed subtype order to place any remainder within a `relation_type`.
- Within these quantity constraints, choose the most natural preference-subtype pairing.
- Prefer pairings that can produce faithful, natural cases with the least unsupported invention.
- Do not invent synthetic facts, topics, task labels, or unsupported conflicts.

"Think step by step. Return only JSON: {`planned_relations`*: [...]}."*

Figure 8: User-related relation-planning prompt for assigning compatibility relation type and subtype before variant generation.

- Each query receives three correct and three incorrect answer candidates, with relation-specific rules for integrated complementary answers, context-specific answers, dated temporal answers, and contradiction-aware answers.

- Separate filters validate session embedding, final query quality, and answer candidates.

Figures 26 and 27 show the query and answer-generation prompts, Figures 28, 29, and 30 show the filters, and Figure 25 shows a real evaluation instance from the same case.

C.5 Stage 5: User-history Assembly

During final assembly, non-user sessions are selected after filtering against the full pool of user-related variant facts, which removes externally sourced facts that are too close to persona-specific memories. The remaining non-user variant sets are assigned to personas without reuse and balanced by compatibility relation type. Their timestamps are redistributed within each persona's user-related time span, or within a fixed date range when no span is available. Each release unit contains a chronological session history and a benchmark file. The system receives only the history and user request, not semantic variants, relation labels, or answer candidates.

C.6 Final data composition.

Across all construction filters, the final pass rate is 88.86% for user-related candidate semantic variant sets and 73.23% for non-user candidate semantic variant sets. Table 7 reports the source and compatibility-relation composition behind the main-text scale summary, and Table 8 reports the domain distribution of semantic variant sets.

C.7 Human Validation of Benchmark Instances

We conduct a human audit to assess whether the constructed benchmark instances preserve their intended evidence, relations, and reference answers.

Sampling and annotation. We sample 150 evaluation instances stratified by relation type, including 50 Complementary, 50 Nuanced, and 50 Contradictory instances. Two annotators independently review each instance without access to its predefined relation label or subtype. For each instance, the annotators examine the interaction history, target semantic variants, resolution target, evaluation query, and reference answer.

Each annotator performs the following four steps:

1. **Evidence recovery.** Locate each target semantic variant in the interaction history and record its supporting session IDs and corresponding evidence.

User-related variant-generation prompt excerpt

"Given the persona and one source preference, write one natural memory case using the required relation type and subtype."
"Persona: ... Topic preference: ... Source preference: ... "
"Relation type definitions:"

- `complementary`: Multiple semantic variants are related to the same question or topic and do not conflict, but they may differ in whether they determine the answer.
- `nuanced`: Multiple semantic variants are relevant to the same question and can all be true, but only under different qualifying conditions such as time, role, location, task, scope, version, definition, or attribute.
- `contradictory`: Multiple semantic variants cannot all be true under the same interpretation and conditions, and cannot be reconciled by adding qualifiers.

"Boundary rules: ... "
"Required relation: {..., `relation_type_meaning`*: ...,* `relation_subtype_meaning`*: ... }."*
"Requirements:"

- Use exactly this relation type and subtype for the case.
- The case must still be natural, faithful to the persona/source preference, and correctly labeled.
- Label *Temporal* or *Contextual* only as nuanced.
- If multiple facts simply support the same stable answer, choose complementary *Multi-evidence* instead of nuanced *Contextual*.
- For *Multi-evidence* cases, write facts that a plausible single user task must combine. Each fact should add a different required constraint, criterion, audience, setting, or scope; avoid facts that are just repeated examples of the same preference.
- For *Contextual* cases, write facts so the same practical user task would require different concrete actions, priorities, or exclusions under each context. Avoid cases where both facts merely show the same general trait.
- *[Condensed from original prompt:]* Write one concise case description and precise one-sentence facts; keep the case faithful to the persona/source preference and do not mention labels, task framing, or this instruction. ...

"[Condensed from original prompt:] JSON output schema omitted; the original output returns relation labels, relation-choice reason, case description, and facts. ... "

Figure 9: User-related variant-generation prompt for converting one persona preference into a target-conditioned semantic variant set.

User-related variant-filter prompt excerpt

"Review this user semantic variant set for data quality."
"Persona: ... "
"Relation type definitions: ... "
"Boundary rules: ... "
"Case: ... "

- Accept only if the case is natural, faithful to the persona/source preference, correctly labeled, non-duplicative, and internally valid.
- Reject if it is unnatural, mislabeled, factually unsupported, duplicated, conflicting outside the contradictory subtype, or impossible to resolve under its stated relation.
- Do not reject solely because the case shares a broad or background persona fact with another case when the answer-driving facts, user-facing intent, and relation behavior are otherwise distinct.
- Reject *Temporal* or *Contextual* cases if the semantic variants are only additive evidence for one stable answer and no time/context condition changes the answer.
- Reject *Multi-evidence* cases if any required fact can be dropped without changing the concrete answer, or if the facts merely repeat the same broad preference in nearby contexts.
- Reject *Contextual* cases if the listed contexts are different on the surface but do not produce meaningfully different user-facing answers, actions, or constraints.
- Reject *Contextual* cases if both facts mainly express the same broad preference/value and do not force different concrete actions, priorities, or exclusions for a plausible user task.
- Reject *Temporal* cases if the time boundary is not expressed in the case facts themselves.
- For contradictory-relation cases, reject if the semantic variants can all be true together, such as following both African and non-African teams.

"Think step by step."
"Output"
"Return only JSON: {`final_decision`*: ...,* `accepted`*: ...,* `reason`*: ...,* `reject_categories`*: [...]}."*

Figure 10: User-related variant-filter prompt for checking factual support, compatibility-relation fidelity, and boundary conditions.

User-related semantic variant sets

Complementary, Multi-evidence.
Case. Amara would be especially drawn to a community program that mixes live drumming with hands-on science learning.
Semantic variants.

1. Amara enjoys drumming rhythms in community events.
2. She designs interactive STEM workshops for children and teens.
3. She turns physics concepts into playful hands-on activities.

Nuanced, Contextual.
Case. Amara uses different design styles for home spaces and children's science activities.
Semantic variants.

1. At home, Amara prefers minimalist Scandinavian design with clean lines and light wood tones.
2. For children's STEM workshops, she uses bold colors, visual cues, and playful layouts.

Contradictory.
Case. A memory conflict appears around Amara's sustainability habits during grocery shopping.
Semantic variants.

1. Amara prefers reusable eco-friendly shopping bags for grocery trips.
2. Amara says she never brings reusable bags and always uses disposable shopping bags.

Figure 11: Real generated user-related semantic variant sets from SubtleMemory. Each block shows the variants that form one target-conditioned set and define its compatibility relation.

Non-user semantic variant sets

Complementary, Multi-evidence.
Case. The answer requires matching several Korean metropolitan cities with their symbol flowers.
Semantic variants.

1. Busan's symbol flower is Camellia.
2. Incheon's symbol flower is Rose.
3. Ulsan's symbol flower is Pear flower.

Nuanced, Temporal.
Case. The total trophy count for SD Crvena zvezda's clubs changes over time.
Semantic variants.

1. As of 2024-06-01, the clubs had won 854 trophies.
2. As of 2024-08-01, the clubs had won 858 trophies.
3. As of 2024-12-01, the clubs had won 870 trophies.

Contradictory.
Case. The same surface question about *The Fault in Our Stars* can point to different answer scopes.
Semantic variants.

1. Described by intended age group, *The Fault in Our Stars* is young adult.
2. Described by general content, it is realistic fiction.

Figure 12: Real generated non-user semantic variant sets from SubtleMemory. Each block shows the external-knowledge variants that define the compatibility relation.

Non-user complementary fact-selection prompt excerpt

"Rewrite the complementary source data below into a cleaner benchmark-ready complementary fact bundle."
"Goal:"

- Produce one benchmark-ready complementary question.
- Produce one canonical answer for that question.
- Select a manageable fact subset for multi-session generation.

"Hard rules:"

- Keep the result complementary: related facts may distract, support, or redundantly restate the answer, but they must not create contradictory or context-resolved multi-answer behavior.
- Keep the selected fact set manageable: choose between 2 and 6 selected facts total.
- Each selected fact must reference one source memory_id.
- The selected fact roles must obey the subtype definition below.
- The canonical answer should be natural text, not a debug note.

"Subtype definition: . . . "
"Source question: . . . "
"Source answer for hidden grounding: . . . "
"Available source evidence items: . . . "
"Return JSON in this format: . . . "
"Think step by step."

Figure 13: Non-user complementary fact-selection prompt for converting multi-evidence source records into benchmark-ready variant sets.

Non-user contradictory fact-selection prompt excerpt

"Select two source QA entries and rewrite them into one contradictory setup."
"Goal:"

- Create a single canonical conflict question that both selected answers appear to answer under the same apparent condition.
- The result must be contradictory memory, not nuanced memory.
- Remove source qualifiers that would reconcile the answers, such as explicit year, version, edition, role variant, location, scope, or scenario markers.
- The canonical conflict question is an internal benchmark scaffold for later dialogue and QA generation. It may later be reused directly or lightly paraphrased.

"Hard rules:"

- Select exactly two different source QA entries.
- Use only answer content grounded in the selected source entries. Do not invent a new answer.
- The canonical conflict question must not quietly preserve a qualifier that makes both answers true.
- Do not output a question like "as of 2016 vs 2017", "in the 1967 or 2016 version", "for the TV series", or anything else that turns the setup back into temporal/contextual nuance.
- Good outcome: two mutually incompatible answers to one plain question.
- Bad outcome: two answers that are still separated by explicit time/version/context qualifiers.

"Source question: . . . "
"Available source QA pairs: . . . "
"Return JSON in this format: . . . "
"Think step by step."

Figure 14: Non-user contradictory fact-selection prompt for selecting conflicting QA entries and removing resolving qualifiers.

Non-user nuanced fact-selection prompt excerpts

"Rewrite the context source data below into a cleaner context fact bundle."
"Goal:"

- Produce one brief ambiguous context question for the topic.
- Keep the result clearly contextual: the different answers should all remain true, but under different contexts.
- *[Condensed from original prompt:]* Preserve every source fact; rewrite each decisive context cleanly and pair it with the correct answer. . . .

"Hard rules:"

- Keep every source evidence item exactly once. Do not drop, merge, or invent facts.
- Preserve the decisive context for each fact, such as role, location, task, scope, version, definition, unit, or attribute.
- *[Condensed from original prompt:]* Do not collapse facts into a summary fact; keep the broad question ambiguous; make each `context_anchor` short and concrete; write compact fact statements that bind the answer to its context. . . .

"Rewrite the HoH temporal source data below into a cleaner temporal fact bundle."
"Goal:"

- Produce one brief umbrella question for the topic. It should be the same underlying question across all snapshots.
- Keep the result clearly temporal: the answer to the same question changes across dated snapshots.
- *[Condensed from original prompt:]* Preserve every dated snapshot exactly once; keep the snapshot date and answer paired for each source fact. . . .

"Hard rules:"

- Keep every source evidence item exactly once. Do not drop, merge, or invent facts.
- The time condition must include the exact source snapshot date, e.g. "as of 2024-07-01".
- *[Condensed from original prompt:]* Do not turn the snapshot date into the event date; do not collapse snapshots into one summary fact; keep the umbrella question short, natural, and without a time condition. . . .

Figure 15: Non-user nuanced fact-selection prompts for converting contextual and temporal source records into benchmark-ready variant sets.

User-related session-generation prompt excerpt

"Generate one realistic user-assistant session for this user."
"Persona: . . . Fact to embed naturally: . . . Case context: . . . Candidate conversation types: . . . Current case relation guidance: . . . Preferred conversation type: . . . Fallback conversation types: . . . Existing sessions for this same case: . . . Persona signal level: . . . "

- Choose exactly one candidate type and write the session in that style.
- Treat the selected conversation type as the primary agent task, not merely as an output format.
- If a candidate type includes a `suggested_flow`, use it as a loose rhythm for the selected type, not a rigid template; if the preferred type would feel forced, choose the most natural fallback type.
- Use existing sessions only as same-case relation and style context. Vary only expression style, such as conversation type, opening shape, request task, assistant output format, and rhythm. Do not change relation-critical context.
- The target fact is hidden supervision. Do not state it directly, but make it recoverable from the user's side of the full session.
- Before writing, identify the target object/category and any qualifier/context. User turns should provide that object/category plus at least one concrete relation signal, such as polarity, behavior, boundary, example, routine, correction, artifact detail, or use context.
- Only embed the target fact for this session. Other facts in the same case are relation context only. Use them to avoid breaking the case design, but do not express, imply, paraphrase, combine, or leak them in this session.
- *[Condensed from original prompt:]* Hidden does not mean cryptic: the fact should be recoverable, but the session should not copy or closely paraphrase the fact, turn it into a direct self-profile sentence, front-load all background into the opening user message, or leak other same-case facts. . . .
- Apply `persona_signal_level` only to the user's broader voice; assistant messages should respond naturally and should not restate the hidden fact as a conclusion about the user.
- Use OpenAI message format with exactly the requested number of turns and messages.
- Messages must start with the user and alternate user, assistant, user, assistant until the session ends.

"Think step by step."
"Output"
"Return only JSON: {`selected_conversation_type`*: . . . ,* `messages`*: [. . .]}."*

Figure 16: User-related session-generation prompt for embedding one semantic variant into an implicit task-oriented dialogue.

User-related session-filter prompt excerpt

"Review a batch of generated user-assistant conversation cases."
"Relation type definitions: . . . "
"Relation subtype definitions: . . . "
"Subtype notes: . . . "
"Boundary rules: . . . "
"Batch: . . . "

- Find only cases that should be rejected.
- For each conversation, first inspect user turns only and internally extract the target object/category, any qualifier/context, and the recoverable signals that make the target fact inferable.
- Reject as `too_hidden` if user turns reveal only a broad topic/category, or if they lack the target object/category plus at least one concrete relation signal.
- Treat hiddenness as acceptable when the conversation does not state the fact directly, but the target object/category and the user's polarity, behavior, boundary, or practical use are recoverable from user turns.
- Reject as `too_explicit` if the conversation directly repeats the fact, closely paraphrases it, or turns it into a profile sentence.
- Reject a case if any conversation is unnatural, too mechanical, not a realistic multi-turn session, does not start with the user, does not alternate user/assistant, does not naturally and implicitly express its target fact, expresses the target fact only as a vague topic-adjacent aside that a reader could not recover from the full session, copies or closely paraphrases the fact, states the fact as a direct self-profile sentence, mismatches the selected conversation type, conflicts with the persona/case/fact, or exposes benchmark, memory, label, or task framing.
- For non-artifact conversation types, reject conversations whose main task is actually drafting, rewriting, translating, caption writing, or repeated text revision.
- Reject a case if its conversations no longer satisfy the current relation type/subtype design, even if each single conversation looks locally fluent.
- Reject a non-*Temporal* case if it becomes valid only by treating a later conversation as the current truth or an earlier conversation as superseded; reject a *Temporal* case if its time split is only implied by artifact/session order rather than content.
- For every rejected case, list the specific problem conversations and their problems when identifiable.
- If there are no problem cases in this batch, return an empty `problem_cases` list.

"Think step by step."
"Output"
"Return only JSON: {`problem_cases`*: [. . .]}."*

Figure 17: User-related session-filter prompt for checking naturalness, variant recoverability, and compatibility-relation preservation.

User-related session excerpts for a Multi-evidence variant set

Case. Amara would be especially drawn to a community program that mixes live drumming with hands-on science learning.
Target facts. Amara enjoys drumming rhythms in community events, she designs interactive STEM workshops for children and teens, and she turns physics concepts into playful hands-on activities.

Session 1: `workflow_setup`.
User. I need a repeatable prep routine for community events. Something simple I can do the night before and again right before things start.
Assistant. Let's make it short enough to reuse. I would split it into night-before packing, arrival setup, and a one-minute reset before the first group joins.
User. Mostly I am trying not to forget my small drum, water, and the first few call-and-response cues before people gather.
...
User. My hands feel stiff at first, and once the crowd answers back the tempo can speed up before I notice.
Assistant. The checklist should keep the drum, cue order, warmup, and tempo anchor together. ...

Session 2: `troubleshooting`.
User. One of my outreach activities keeps falling apart halfway through, and I am trying to figure out what the actual problem is.
Assistant. Tell me what the activity is, who is in the room, and where the breakdown usually starts.
User. It is a ramp-and-toy-car lesson on speed and angle. I usually have both younger kids and teens together.
...
User. I need it reusable for our STEM programs, including school and museum rooms where groups move at different speeds.
Assistant. Split the activity into a quick shared demo, a younger-kid lane, and a teen extension challenge.
...

Session 3: `resource_selection`.
User. I am choosing one physics activity for a community family pop-up, and I only have room for a single setup.
Assistant. The best choice depends on whether you want the visitors to observe a demonstration or change variables themselves.
User. I want that little sequence of predict, try, argue with reality, and try again.
...
User. Bonus points if it feels like a game and not a school lesson, works on a modest budget, and lets participants touch the system themselves.
Assistant. Choose toy-car ramp trials because they make changing motion visible through repeated hands-on tests. ...

Figure 18: Real user-related session excerpts from a complementary Multi-evidence variant set. The compatible facts are distributed across ordinary task-oriented interactions, and a later community-program query requires recovering all three sessions rather than using any single session alone. Ellipses indicate omitted turns from the original sessions.

Non-user session-planning and session-generation prompt excerpts

Session planning. *"Plan independent user-assistant sessions for a [complementary-relation / nuanced-relation / contradictory-relation] memory benchmark."* The prompt asks the model to produce the required number of session plans, assign selected semantic facts according to the relation-specific rule, make the sessions differ in scenario and concrete event, choose one conversation-type candidate and its paired flow for each session, and keep facts assigned to other sessions unmentioned. *[Condensed from relation-specific prompt variants:] Multi-evidence* cases distribute required facts for cross-session integration; *Any-one* cases keep equivalent facts redundant but natural; *Contextual* and *Temporal* cases preserve the decisive context or snapshot condition; contradictory-relation cases embed exactly one target claim per session while staying silent about contradiction. ...

Session generation. *"Generate one standalone user-assistant session for a [complementary-relation / nuanced-relation / contradictory-relation] benchmark."* The prompt asks for a realistic everyday dialogue that follows the approved plan, uses the chosen conversation type and flow as the interaction rhythm, embeds only the assigned facts, keeps the event distinct from other hidden sessions, avoids asking the benchmark question verbatim, and avoids clean inventories or one-shot recaps. *[Condensed from relation-specific prompt variants:]* Partial *Multi-evidence* sessions must not state the full answer; *Contextual* sessions keep qualifiers explicit enough to matter; *Temporal* sessions treat dates as answer-snapshot conditions; and contradictory sessions do not mention disagreement, mixed memories, or other sessions. ...

Output schema. *"Return JSON in this format:"* `{session_id, chosen_scenario, conversation, coverage_notes}`.

Figure 19: Non-user session-planning and generation prompts for distributing selected external-knowledge variants across natural dialogues.

Non-user session excerpts for a Multi-evidence variant set

Case. The answer requires matching several Korean metropolitan cities with their symbol flowers.
Target facts. Busan's symbol flower is Camellia; Ulsan's symbol flower is Pear flower; Incheon's symbol flower is Rose.

Session 1: `learning_explanation`.

User. I'm helping my younger cousin study Korean city symbols. Can you explain in simple terms what a city's symbol flower is, and how to make the idea easy to remember?
Assistant. A city's symbol flower is an official flower the city uses to represent itself, a bit like a visual emblem. The easiest way to remember it is: not "favorite flower," but "flower that stands for the city."
User. That helps, but I learn better with examples. Could you use one metropolitan city so I can hear how you'd explain it out loud?
Assistant. If I were explaining it with Busan, I'd say that Busan's symbol flower is Camellia. Then I'd add that this is one example of how a city chooses a flower to act as part of its civic identity.
User. Nice. Could you give me a second city too? I think comparing two examples would make the concept stick better.
Assistant. You could pair Busan with Ulsan and point out that Ulsan's symbol flower is Pear flower. That shows the main idea: each city has its own emblematic flower.
User. I like that. When I explain it, should I focus more on the flower name or on what the symbol is doing?
Assistant. Focus first on what the symbol is doing. With Busan, don't stop at "Camellia" by itself; say that Busan uses Camellia as its symbol flower.
...
User. One last thing: can you give me a memory trick for the Ulsan example?
Assistant. Link the word "pear" with the image of a light spring blossom. If you picture Ulsan paired with a pear blossom image, it becomes easier to recall that Ulsan's symbol flower is Pear flower.

Session 2: `artifact_production`.

User. I'm putting together a small cultural display and need a short caption for an Incheon profile card.
Assistant. Here's a clean starting option: "Incheon, a dynamic gateway city, blends modern energy with local pride and tradition."
User. Nice start. I want it to feel a little more specific though, like something that belongs on a city handout card.
Assistant. "Incheon is a vibrant city known for its forward-looking spirit; its symbol flower is the rose, reflecting beauty and civic pride."
User. That's close, but it sounds a bit formal. Can you make it warmer and slightly shorter?
Assistant. "Incheon is a lively city whose symbol flower, the rose, represents its beauty and pride."
...
User. I like that. Could you give me two final options: one plain and one a little more polished? Keep both short.
Assistant. Plain: "Incheon's symbol flower is the rose." Polished: "The rose, Incheon's symbol flower, reflects the city's beauty and pride."
User. Great, I'll use the polished one. Can you give it to me one last time with no extra commentary so I can paste it into the layout?
Assistant. The rose, Incheon's symbol flower, reflects the city's beauty and pride.

Figure 20: Real non-user session excerpts from a complementary Multi-evidence variant set. The required facts are distributed across ordinary task-oriented interactions rather than exposed as a benchmark list.

User-related query-generation prompt excerpt

"Generate candidate user queries for this user semantic variant set."
"Persona: … Case: … Case relation: … Accepted sessions for this case: … Same-topic sibling cases for interference avoidance: … Relation-specific task guidance: … "

- Generate exactly the requested number of natural first-person user task requests.
- Write each request from the user's point of view as a natural user-to-agent help request, not as a direct question about the user's own preference, habit, usual choice, or best fit.
- Each request must be self-contained, realistic, have one primary decision/output target, and match the current case relation definition.
- A correct response must require the current case facts and relation; avoid broad or generic tasks that can be completed well without this case.
- Prefer task shapes whose correctness can be checked through concrete selection, exclusion, ranking, checklist items, clarification, or fact-based criteria.
- Include only neutral time/context/role/location/task/scope/version/object conditions needed by the relation; leave the selected preference, habit, conflict, or state unstated.
- *[Condensed from original prompt:]* Use accepted sessions only as background signal; do not refer to prior chats, hidden facts, ground-truth preferences, memory, labels, benchmark framing, answer direction, or details that appear only in sessions. …

"Task form menu:" "[Condensed from original prompt:] Choose exactly one `task_form`*. Each query must include a visible user goal, agent-available material, and a concrete output contract.* `structured_form` *uses fixed fields or slots that make the answer judgeable;* `resource_arrangement` *uses visible resources and asks for selection, ranking, exclusion, assignment, scheduling, cart building, or a similar concrete operation. Use realistic named resources, matched distractors, and enough visible ambiguity that memory is needed. … "*
"Hint and interference control:" "[Condensed from original prompt:] The query must be underdetermined without the current case facts. Visible goals, fields, resources, and neutral context may make the task natural, but they must not reveal the correct answer. Avoid option-specific hints, hidden-criterion names, visible answer direction, and same-topic sibling interference; run a visible-only check and rewrite if the query alone would make the answer obvious. … "
"[Condensed from original prompt:] Task quality calibration examples and JSON output schema are omitted; the original output stores each generated query in the `question` *field together with its* `task_form`*. … "*

Figure 21: User-related query-generation prompt for producing target queries q_τ in assistant-task form.

User-related answer-generation prompt excerpt

"Generate short fact-grounded answer candidates for these user queries."
"Persona: ... "
"Case: ... "
"Case relation: ... "
"Accepted sessions for this case: ... "
"Queries: ... "
"Relation-specific answer guidance: ... "

- For each query identifier, generate exactly the requested number of correct answers and incorrect answers.
- Treat each item as a concise, standalone key answer, not as the full task output.
- Answers must match the current case relation, respect stated time/context conditions, and be grounded in the case facts and persona rather than unseen prior text.
- Correct answers should capture the essential facts, fact combinations, conditions, directions, exclusions, or candidate decisions a good response must reflect.
- Use `task_form` when present: `structured_form` keys identify required field values or slot content, while `resource_arrangement` keys identify selected or excluded resources, ordering, final set, slot assignment, needed clarification, or abstention/no-selection when the relation requires it.
- Incorrect answers should be plausible but wrong: wrong facts, missing facts, wrong condition/direction, wrong form values, wrong candidate/order/final set, surface-clue-only choice, stereotype-driven choice, overconfident contradiction resolution, hedged non-selection, or unsupported criterion.
- If external named candidates appear, evaluate the in-prompt, case-grounded candidate decision rather than trivia or outside-world facts.
- Keep answers natural, specific, similar in length, and limited to the key answer content.
- Do not write meta-evaluative statements, full deliverables, unsupported persona comparisons, or mentions of labels, memory, benchmark framing, hidden facts, hidden preferences, or task framing.

"[Condensed from original prompt:] JSON output schema omitted; the original output uses `question_id` *as the query identifier and returns* `correct_answers` *and* `incorrect_answers`*. ... "*

Figure 22: User-related answer-candidate generation prompt for producing reference correct answers A^+ and plausible incorrect answers A^-.

User-related instance-filter prompt excerpt

Task. Review generated user queries and short answer candidates.
Input.

- Batch: ...
- Each case QA item includes `case_relation_guidance`. Use the current relation type meaning, subtype meaning, and subtype notes when judging relation quality.

Reject query.

- *[Condensed from original prompt:]* Reject queries with unnatural or overly broad requests, no realistic completion need, multiple deliverables, dependency on prior sessions or drafts, direct preference queries, benchmark/memory leakage, hidden-fact paraphrase, relation mismatch, or tasks that mainly reward generic writing rather than fact use. ...
- *[Condensed from original prompt:]* Reject task-form failures such as missing or invalid `task_form`, missing visible goal, agent-available material, or output contract, uncontrolled form fields, missing resource briefs, requests to invent/search/rewrite resources, over-tuned candidates, option-specific hints, weak distractors, visible-query answerability, stereotype answerability, or sibling-case interference. ...

Relation checks.

- *[Condensed from original prompt:]* For complementary-relation cases, reject invented conditions or conflicts; *Multi-evidence* answers must combine required memories.
- For nuanced-relation cases, reject missing or exposed conditions, wrong conditions, global flattening, and tasks without real temporal/contextual discrimination.
- For contradictory-relation cases, reject one-side or invented-resolution prompts; correct answers must acknowledge conflict or uncertainty.
- *[Condensed from original prompt:]* Rejection calibration examples cover `missing_or_invalid_task_form`, `missing_external_candidate_framing`, `over_tuned_candidates`, `case_keyword_leakage`, `option_specific_hint`, `answerable_without_memory`, and weak *Contextual*. ...

Answer checks.

- *[Condensed from original prompt:]* Remove correct answers that are unsupported, too generic, condition-insensitive, contradiction-resolving, or written as full task outputs.
- Remove incorrect answers that are actually correct, contain the target fact, reward the right condition, or remain valid answer targets.
- If a query has no valid correct answers or no valid incorrect answers, reject that query.

Output. *[Condensed from original prompt:]* JSON schema omitted; the original output returns rejected queries and removed answers. ...

Figure 23: User-related instance-filter prompt for validating generated target queries q_τ and answer candidates.

User-related evaluation instance excerpts for two task forms

Case. Amara uses different design styles depending on whether she is shaping her home environment or creating materials and spaces for children's science activities.
Facts.

1. At home, Amara prefers minimalist, Scandinavian-inspired design with clean lines, light wood tones, and an uncluttered feel.
2. For children's STEM workshops or toy-car race setups, she uses bold colors, interactive visual cues, and playful layouts.

Task form: `structured_form`.
User request. Fill a one-card apartment design brief with short phrases: overall direction, shape/line cue, wood/finish cue, clutter boundary, and room feeling.
Reference correct answer. Minimalist Scandinavian; clean lines; light wood tones; uncluttered surfaces; calm ease.
Plausible incorrect behavior. Use the workshop style instead: bold graphic forms, vivid finishes, layered visual cues, and energetic engagement.

Task form: `resource_arrangement`.
User request. Choose one visual setup package for an elementary museum toy-car physics activity from a provided resource brief.
Reference correct answer. Signal Sprint Kit – it matches bold colors, interactive cues, and small activity islands for an energetic kids setup.
Plausible incorrect behavior. Choose Vector Grove Kit because its pale wood and restrained palette match the home-design preference instead.

Figure 24: User-related evaluation-instance excerpts showing the two task forms used in user-related query construction. Both examples come from the same nuanced Contextual case: the correct response depends on applying the appropriate remembered context rather than a superficially similar one.

Non-user evaluation instance excerpt for a Multi-evidence variant set

Target query. Before I finish my notes, can you remind me which symbol flower belongs to Busan, Incheon, and Ulsan?
Reference correct answer. Busan's symbol flower is Camellia, Incheon's is Rose, and Ulsan's is Pear flower.
Plausible incorrect behavior. Busan's symbol flower is Camellia, Incheon's is Pear flower, and Ulsan's is Rose.
Required evidence use. The answer must combine facts distributed across the two sessions: Busan and Ulsan are mentioned in the learning explanation session, while Incheon is mentioned in the artifact-production session.

Figure 25: Real non-user evaluation-instance excerpt for the same Multi-evidence variant set shown in Figure 20. The target query is answerable only if the system recovers and combines the relevant facts across sessions.

Non-user query-generation prompt excerpts

Complementary. *"Write one new follow-up question based on the sessions below."*

- Produce exactly one question that is answerable from the stored sessions alone, sounds like a natural later follow-up from the same user, is self-contained, and does not become a meta-question about memory, sessions, or process.
- For *Multi-evidence*, preserve the hidden dependency structure instead of restating every hop in the question. Do not enumerate all intermediate facts, titles, locations, or aliases; use light anaphora or natural references when possible.
- For MuSiQue-derived chains, avoid riddle-like nested clauses and preserve the benchmark difficulty through recalled session context and the final reasoning relation rather than through over-engineered surface wording.

Nuanced Contextual. *"Write one or two new follow-up questions based on the sessions below."*

- Every question must specify the decisive context and target exactly one context fact. The answer should be a single context-specific answer, not a full list of all contexts.
- If two questions are produced, they must target different memory identifiers and ask from meaningfully different angles.
- Each question must remain understandable after many unrelated sessions have been stored. Include enough topic, entity, and context clues, such as role, location, jurisdiction, scope, version, definition, unit, attribute, object range, or scenario.
- Do not ask the broad ambiguous question directly, ask for all contexts, or use vague local references such as "what is it in that case?"

Nuanced Temporal. *"Write exactly 3 new follow-up questions based on the sessions below."*

- Produce exactly three question objects, each asking for the answer to the same underlying question under one temporal condition and resolving to exactly one target memory identifier.
- Cover exactly one easy, one medium, and one hard question. Easy questions may name a direct as-of date; medium questions should use a natural event, task, draft, note, version, or session anchor; hard questions should use relative temporal reasoning or a latest-before anchor.
- Include enough topic, entity, fact-property, and temporal or task clues for a memory system to locate the intended snapshot.
- Do not ask for all dates, all periods, a full timeline, chronological order, or the current/latest answer without a temporal anchor.

Contradictory. *"Write one new question based on the independent sessions below."*

- Produce exactly one self-contained, natural question that targets the main disputed point.
- Do not mention "contradiction," "conflict," or "different sessions," and do not add qualifiers that would resolve the contradiction.
- Start from the canonical conflict question. It may be reused directly or lightly paraphrased, but the paraphrase must not choose a side or reintroduce time, version, edition, role, scope, or context qualifiers that make both claims compatible.

Output schema. *"Return JSON in this format:"* `{question}` for complementary-relation and contradictory-relation cases, and `{questions: [{target_memory_id, question, question_design_notes, ...}]}` for nuanced-relation cases.

Figure 26: Non-user query-generation prompts for producing target queries q_τ from accepted external-knowledge sessions. The box summarizes the relation-specific prompt variants for complementary-relation, nuanced-relation Contextual, nuanced-relation Temporal, and contradictory-relation cases.

Non-user answer-generation prompt excerpts

Shared instruction. *"Generate answer candidates for the question below."*

- Produce exactly three correct answers and exactly three incorrect answers.
- Each answer must be a dictionary with one key, `text`.
- Keep all six answers plausible, natural, concise, and reasonably similar in style and length without repeating the same wording.

Complementary.

- Every correct answer must mean the same thing as the canonical answer.
- If the canonical answer is structured, preserve all entities, pairings, and key details completely.
- Incorrect answers should prefer distractor-grounded local confusions instead of arbitrary unrelated alternatives.

Nuanced Contextual.

- Every correct answer must be the single answer for the target context only.
- Correct answers may briefly restate the context, but must not expand into all context-conditioned answers.
- Incorrect answers should reflect realistic context confusions, such as choosing the answer from another context for the same broad topic; do not use random unrelated wrong answers.

Nuanced Temporal.

- Every correct answer must mean the same thing as the target answer for the requested temporal snapshot.
- Correct answers should preserve key entities, numbers, and names, and may briefly mention the temporal anchor, but must not expand into the whole timeline.
- Incorrect answers should reflect realistic temporal confusions, such as choosing an answer from a different snapshot date for the same underlying question.

Contradictory.

- Every correct answer must explicitly state that the remembered sessions conflict or remain unresolved.
- Correct answers must not choose a side, pretend one claim is final, or quietly repair the contradiction.
- Good correct answers may name the competing claims and say that clarification or verification is needed.
- Incorrect answers should reflect failure modes such as picking one side, pretending one session overrules the other, inventing a clean resolution, or ignoring the practical framing of the question.

Output schema. *"Return JSON in this format:"* `{correct_answers: [{text}, {text}, {text}], incorrect_answers: [{text}, {text}, {text}]}`.

Figure 27: Non-user answer-candidate generation prompts for producing reference correct answers A^+ and plausible incorrect answers A^- under each compatibility relation type.

Non-user conversation-filter prompt excerpts

"You are judging one [complementary/nuanced/contradictory]-memory benchmark sample. Return valid JSON only and follow the requested schema exactly. Judge strictly against the subtype requirements and naturalness requirements. Think step by step."
"Complementary conversation-stage judging requirements:"

- Judge only the conversation quality, not the final benchmark question or answer candidates.
- The multi-session conversations should feel like ordinary user-assistant interactions, not quiz reveals, benchmark templates, or explicit fact dumps.
- *[Condensed from original prompt:]* Accept varied task scenarios; reject sessions that fail the complementary subtype, leak the target too directly, or integrate source facts unnaturally. . . .

"Nuanced conversation-stage judging requirements:"

- Judge only the sessions and how the nuanced facts are embedded.
- Sessions should feel natural, distribute facts across interactions, and avoid explicit benchmark recaps.
- *[Condensed from original prompt:]* For Temporal, sessions must embed as-of/snapshot/version answers for the same underlying question, not separate event dates; for Contextual, facts must remain valid under different conditions. Reject samples that become contradiction, separate-entity facts, or explicit recaps. . . .

"Contradictory conversation-stage judging requirements:"

- Judge only the sessions and how the contradictory facts are embedded.
- Sessions should feel independent and ordinary, with the contradiction emerging across sessions rather than through explicit debate.
- *[Condensed from original prompt:]* Reject sessions that openly announce, debate, choose a side, or resolve the contradiction, or that make the facts separable by time, version, or context. . . .

"Return JSON in this format: . . . "

Figure 28: Non-user conversation-filter prompts for validating generated sessions before target-query construction.

Non-user question-filter prompt excerpts

"You are judging the question quality of one [complementary/nuanced/contradictory]-memory benchmark sample. Return valid JSON only and follow the requested schema exactly. Judge strictly against the subtype requirements and question-design requirements. Think step by step."
"Complementary question-stage judging requirements:"

- Judge only the final benchmark question, with access to the sessions for context.
- Do not judge the answer candidates here.
- The question should sound like a natural later follow-up and fit the subtype: *Multi-evidence* requires integrating multiple facts, while *Any-one* targets a shared answer recoverable from any equivalent mention.
- *[Condensed from original prompt:]* Reject questions that are too direct, unnatural, subtype-mismatched, or answerable in the wrong way. . . .

"MuSiQue-specific judging adjustment:" "[Condensed from original prompt:] MuSiQue questions may use concrete names, short labels, or session anchors; long chains are allowed; do not mechanically enforce the original effective_k count if the question still requires meaningful memory-grounded synthesis. . . . "
"Nuanced question-stage judging requirements:"

- The question should sound natural and subtype-appropriate.
- *[Condensed from original prompt:]* For Temporal, generate exactly three questions with different angles and difficulty levels; each must resolve to one dated/as-of/version snapshot for the same underlying question and should not ask for a timeline, all dates, latest/current without an anchor, or a full recap. . . .
- *[Condensed from original prompt:]* For Contextual, specify one decisive context such as role, location, scope, version, definition, unit, attribute, jurisdiction, or scenario; reject broad ambiguous questions, full lists, all-context questions, or questions answerable by more than one selected context fact. . . .

"Contradictory question-stage judging requirements:"

- The question should directly ask about the disputed fact.
- *[Condensed from original prompt:]* Reject questions that quietly add resolving time/version/context qualifiers, turn into meta-questions about memory or disagreement, or fail to target the disputed point. . . .

"Return JSON in this format: . . . "

Figure 29: Non-user question-filter prompts for validating target queries under complementary, nuanced, and contradictory relation requirements.

Non-user answer-filter prompt excerpts

"You are judging the answer quality of one [complementary/nuanced/contradictory]-memory benchmark sample. Return valid JSON only and follow the requested schema exactly. Judge strictly against the subtype requirements and answer-design requirements. Think step by step."
"Complementary answer-stage judging requirements:"

- Judge only the answer candidates.
- The three correct answers should all be genuinely correct for the question and semantically aligned with the intended canonical answer.
- The answers should match the subtype: *Multi-evidence* preserves the integrated multi-fact result, while *Any-one* targets the shared answer.
- *[Condensed from original prompt:]* Incorrect answers are auxiliary distractor metadata; do not reject merely because they are easy or weak, but reject actually correct/equivalent distractors, ambiguous answer sets, wrong correct answers, incomplete correct answers, or subtype-mismatched correct answers. ...

"Nuanced answer-stage judging requirements:"

- All three correct answers must be genuinely correct for the subtype and question.
- For Temporal, all correct answers should match only the target dated/as-of/version snapshot. If the question uses an event/version/relative anchor rather than an explicit date, use the sessions and `target_memory_id` metadata to judge the intended snapshot.
- For Contextual, all correct answers should match only the context explicitly specified by the question.
- *[Condensed from original prompt:]* Incorrect answers are auxiliary distractor metadata; reject actually valid/equivalent distractors, ambiguous answer sets, incomplete or wrong correct answers, all-context answers, clarification answers, wrong-context answers, or subtype-mismatched correct answers. ...

"Contradictory answer-stage judging requirements:"

- All correct answers must explicitly acknowledge the contradiction, uncertainty, or need for clarification/verification.
- Correct answers must not choose one side as final.
- *[Condensed from original prompt:]* Incorrect answers are auxiliary distractor metadata; do not reject merely because they are easy or repetitive, but reject valid contradiction-aware distractors, ambiguous answer sets, correct answers that pick one side, or correct answers that fail to acknowledge contradiction/uncertainty. ...

"Return JSON in this format: ... "

Figure 30: Non-user answer-filter prompts for validating reference correct answers A^+ and plausible incorrect answers A^-.

Measure	Value	Proportion
Dataset scale		
Persona-level histories	10	–
Memory-bearing sessions	2,364	–
Total session tokens	∼2.12M	–
Avg. tokens per history	∼211.6K	–
Evaluation instances	1,522	–
Variant sets by source		
Semantic variant sets	1,090	100.00%
User-related sets	545	50.00%
Non-user sets	545	50.00%
Variant sets by relation		
Semantic variant sets	1,090	100.00%
Complementary sets	361	33.12%
Nuanced sets	352	32.29%
Contradictory sets	377	34.59%
Instances by source		
Evaluation instances	1,522	100.00%
User-related instances	707	46.45%
Non-user instances	815	53.55%
Instances by relation		
Evaluation instances	1,522	100.00%
Complementary instances	361	23.72%
Nuanced instances	784	51.51%
Contradictory instances	377	24.77%

Table 7: SubtleMemory data composition. Proportions are computed within each dashed group. Token counts are approximate session-token counts.

Domain	Variant sets	Proportion
Culture	191	17.52%
Media	188	17.25%
Competition	147	13.49%
World	143	13.12%
Society	123	11.28%
Cuisine	67	6.15%
Lifestyle	65	5.96%
Development	59	5.41%
STEM	55	5.05%
Nature	52	4.77%
Total	1,090	100.00%

Table 8: Topic-domain distribution of SubtleMemory semantic variant sets. Counts and proportions are computed at the set level.

Validation metric	Average	Agreement
Evidence recoverability	99.7%	99.3%
Relation-label correctness	96.6%	95.9%
Contradictory validity	93.8%	91.7%
Answer-key correctness	99.7%	99.3%
Overall validated	95.9%	94.5%

Table 9: Human validation results on 150 stratified benchmark instances. Average validation rates are computed across the two annotators.

2. **Relation labeling.** Infer whether the target variants are Complementary, Nuanced, or Contradictory based on the variants and their shared resolution target.

3. **Contradiction checking.** For instances identified as Contradictory, determine whether the variants are mutually incompatible under the resolution target or can be reconciled through temporal, contextual, scope, version, or role differences.

4. **Answer-key verification.** Determine whether the reference answer is supported by the recovered evidence and correctly handles the inferred relation.

Validation metrics. We derive four metrics from the annotations. *Evidence recoverability* requires every target semantic variant to be grounded in specific supporting sessions. *Relation-label correctness* requires the relation inferred by the annotator to match the predefined benchmark relation. *Contradictory validity* requires the conflicting variants to remain irreconcilable under the resolution target. *Answer-key correctness* requires the reference answer to be supported by the interaction history and consistent with the inferred relation. An instance is counted as *overall validated* only when it passes all applicable checks.

For each metric, we report the mean validation rate across the two annotators over all applicable annotations, with disagreements retained in the denominator. Inter-annotator agreement is the proportion of jointly applicable annotations on which the two annotators provide the same judgment.

Results. As shown in Table 9, the audit yields a 95.9% overall validation rate with 94.5% inter-annotator agreement. Evidence recoverability and answer-key correctness both reach 99.7%, while relation-label correctness and Contradictory validity reach 96.6% and 93.8%, respectively. These

Relation	Query only	Oracle	Δ (pp)
Complementary	3.37	91.57	88.20
Nuanced	0.88	98.46	97.59
Contradictory	0.00	91.71	91.71
Overall	1.23	95.46	94.23

Table 10: Accuracy (%) on the 815 non-user instances using GPT-5.4. Both settings use the same answer prompt and differ in whether the annotated interaction-history evidence is provided.

results provide quantitative support for the internal validity of the audited benchmark instances. The audit does not measure conversational naturalness or provide an end-to-end human-performance ceiling.

C.8 Target-Dependence Validation

To verify that the non-user evaluation instances require information from the interaction history rather than parametric knowledge alone, we evaluate GPT-5.4 on all 815 non-user instances using only the query, without providing the interaction history or retrieved memory. We use the same answer prompt as in the Oracle setting, where the model receives the annotated target sessions.

As shown in Table 10, we obtain 1.23% overall accuracy using only the query, compared with 95.46% when providing the annotated history evidence. The large gap across all three relation types indicates that the evaluated non-user instances depend on interaction-history evidence rather than parametric knowledge alone.

C.9 Artifact Availability, Licenses, and Intended Use

Code and data are publicly available under the Apache License 2.0. The code repository contains the construction and evaluation pipelines, prompts, and reproduction configurations, while the data repository contains the generated histories and evaluation instances.

Existing seed resources remain governed by their original licenses and access terms. We use them only as seed sources for constructing synthetic benchmark histories and evaluation instances, cite the original resources, and do not redistribute the original source artifacts as standalone data unless their terms permit redistribution. SubtleMemory artifacts are intended for research evaluation of long-term memory agents, not for profiling real individuals or deploying personalized assistants from the synthetic histories. The released artifacts should be used consistently with this research-evaluation purpose and with any source-specific access conditions.

D Experiments

D.1 Experimental Setup

D.1.1 Evaluation-time Baseline and Model Settings

In the actual evaluation, each memory system is run through its native write and retrieval interfaces using the default or author-recommended configuration reported in Table 11. Systems retain their native chunking, storage, retrieval, and parameter settings; no system is tuned on a common development set, and storage and inference budgets are not matched across systems. The results therefore compare the end-to-end memory mechanisms under their evaluated configurations rather than individual components or cost-matched efficiency.

Answer generation is separated from memory construction for systems that expose retrieved context to the evaluation pipeline, while native agent settings let the framework run its own answer loop and prompt injection. We use the GPT-family models in Table 12 for answer generation, and use Gemini 3.1 Pro Preview Thinking as the LLM judge, which compares each generated answer against accepted correct references, known incorrect references, and case metadata. Table 12 summarizes the model sources and key hyperparameters used in the experiments.

D.1.2 Context Organization with OpenClaw

Table 11 distinguishes standalone memory systems from OpenClaw-based agent deployments, but this difference is not only a change in the memory backend. As discussed in the main results around Table 3, adding OpenClaw can change performance even when the external memory system and the target query remain comparable. The reason is that OpenClaw (OpenClaw, 2026) changes the answer-time context organization: standalone memory systems serialize retrieved items directly into the benchmark answer prompt, while OpenClaw-based systems route the same query through an agent workspace whose instructions, plugin recall, and current user turn are organized by the runtime.

Figure 31 gives an artifact-level example using the same Mem0 book-selection query. The example shows what "context organization" means

Baseline	System source	Configurable hyperparameters	Memory-stage model
Mem0	Platform API.	search.top_k=20; max_content_length=8000.	Platform default add model.
MemOS	Platform API.	search.top_k=20.	Platform default add model.
EverMemOS	Platform API.	search.top_k=10; retrieve_method=hybrid; memory_types={episodic_memory}.	Platform default add model.
A-Mem	Local GitHub implementation.	search.top_k=20; retriever=all-MiniLM-L6-v2; evo_threshold=100.	Add model=gpt-4o-mini.
MIRIX	Local GitHub implementation.	search.top_k=20; embedding model=text-embedding-3-small.	Add/search control=gpt-4o-mini; final answer uses configured answer model.
MemoBase	Self-hosted/local API service.	search.backend=context; top_k=20; max context=3000 tokens; event thresholds=0.2.	Add model=gpt-4o-mini.
OpenClaw	Local OpenClaw runtime.	search.top_k=50.	No LLM write model.
MetaClaw	Local MetaClaw runtime.	retrieval_mode=hybrid; top_k=8; max injected memory tokens=2400; max context chars=18000.	No separate write model configured.
OpenClaw-style plugins	Local OpenClaw runtime plus backend plugin.	search_top_k=50; MemOS memory/preference limits=20; EverMemOS retrieval=hybrid episodic.	Backend write model follows the selected plugin.
Oracle context	Local oracle adapter.	None.	No write-stage model.

Table 11: Evaluation-time baseline settings. The table lists the implementation source, fixed memory hyperparameters, and memory-stage model configuration used for each evaluated baseline.

Model	Parameter setting	Source	URL
GPT-4o mini	temperature=0.0, max tokens=16,384	gpt-4o-mini	https://platform.openai.com
GPT-5.4	temperature=0.0, max tokens=16,384	gpt-5.4	https://platform.openai.com
GPT-OSS 120B	temperature=0.0, max tokens=16,384	openai/gpt-oss-120b	https://platform.openai.com
Gemini 3.1 Pro Preview Thinking	temperature=0.0	gemini-3.1-pro-preview-thinking	https://ai.google.dev

Table 12: Answer-generation and judge model settings, including decoding parameters, model identifiers, and provider URLs.

concretely: without OpenClaw, the answer model receives a flat numbered memory list inside the # CONTEXT field of the benchmark prompt; with OpenClaw, the answer rules are loaded as workspace instructions and the Mem0 plugin recall is injected as agent runtime context before the target query. This illustrates why OpenClaw integration should be interpreted as an agent-context intervention, not simply as another formatting of the same retrieved list.

D.1.3 LLM-as-judge Validation

The judge prompt and validation sample are specified as follows:

- The judge receives the target query, generated answer, accepted correct answers A^+, known incorrect answers A^-, supporting facts, case description, relation metadata, source, and relation-specific grading guidance.
- The output is a binary correctness label with a short reason. Grading is semantic rather than surface-form based.
- For validation, we use 45 fixed-seed queries from one of the ten user-history splits, with 15 from each relation type and five candidate answers per query, yielding 225 human-labeled answers.
- Human annotators receive the same information as the judge. The automatic labels reach Cohen's $\kappa = 0.963$ against human labels.

Figure 32 summarizes the judge prompt used for binary answer evaluation.

D.1.4 Answer-generation Prompts

We use the same answer-generation policy across evaluated systems. For each target query, the answer model receives the evidence exposed by the evaluated setting as {context} and the target query as {question}. The answer prompt does not reveal the hidden semantic variant set, relation label, accepted correct answers, or known incorrect answers. Following the main text, we compare a *soft prompt* with general guidance and a *strong prompt* with explicit instructions for target identification, conflict recognition, evidence fidelity, and clarification. Figures 33 and 34 show the two answer-generation prompts.

Answer-context organization with and without OpenClaw

Example setting. We compare one `gpt-5.4` Mem0 item from the complementary book-selection split. Both runs receive the same target query: choose two books from a hold shelf containing *So Long a Letter*, *The Concubine*, *Passing*, *Open Water*, *Homegoing*, and *A Month in the Country*. The reference answer selects *So Long a Letter* and *The Concubine* because both match the remembered preference for African literary classics.

Without OpenClaw: standalone Mem0. The evaluation pipeline directly assembles one answer prompt. The prompt contains the benchmark answer rules, then a numbered `# CONTEXT` block built from Mem0 retrieval, and then the target query. The saved context for this item begins as follows:

> *1. 12:10 pm on 21 April, 2025: Assistant recommended So Long a Letter as the gliding novel and Nervous Conditions as the push-back choice, noting Efuru as a softer backup if needed.*
>
> *2. 12:08 pm on 21 April, 2025: Assistant recommended five compact literary classics: So Long a Letter, Efuru, Nervous Conditions, The Concubine, and Weep Not, Child.*
>
> *3. 12:08 pm on 21 April, 2025: User prefers the "compact literary classics" lane, gravitating toward works similar to Chinua Achebe and Buchi Emecheta, and wants books between 180 and 300 pages.*
>
> *4. 12:10 pm on 21 April, 2025: Assistant ranked the five suggested compact literary classics: So Long a Letter first, Efuru second, Nervous Conditions third, The Concubine fourth, and Weep Not, Child last.*

The answer model therefore sees the benchmark instructions, the retrieved memory list, and the query in one benchmark-formatted prompt.

With OpenClaw: Mem0 + OpenClaw. The target query is passed to the OpenClaw local agent as the current user turn. The answer rules are loaded from the OpenClaw workspace instructions, and the Mem0 plugin supplies recalled memory as OpenClaw runtime context before the user turn. The corresponding runtime-injected context begins as follows:

> *OpenClaw workspace instruction excerpt:*
>
> - *This is a SubtleMemory memory-system evaluation workspace.*
> - *Answer the current benchmark question using only the current question and OpenClaw recall-injected memory context.*
> - *Recalled memory/context may be auto-recalled or available through the configured OpenClaw memory backend or plugin.*
> - *Do not use outside knowledge, browser/web search, or unrelated workspace files. Keep the final answer concise and grounded in the recalled memory context.*
> - *SubtleMemory Answer Rules: (1) Use only the recalled memory/context injected before the question. (2) Your first priority is evidence fidelity: detect true unresolved conflicts, but do not over-detect conflicts from compatible evidence. (3) Identify the exact target needed by the question . . .*
>
> *1. 12:10 pm on 21 April, 2025: Assistant ranked the five compact literary classics, naming So Long a Letter as the best first pick, Efuru as steady and lucid, Nervous Conditions as the edgier option, The Concubine as more atmospheric, and advised saving Weep Not, Child for a later weekend.*
>
> *2. 12:08 pm on 21 April, 2025: Assistant recommended five compact literary classics fitting the 180–300 page range: So Long a Letter, Efuru, Nervous Conditions, The Concubine, and Weep Not, Child.*
>
> *3. 12:08 pm on 21 April, 2025: User requests that at least one suggested novel be authored by a woman, prefers clean prose over stylistic fireworks, and wants to avoid emotionally heavy reading.*
>
> *4. 12:08 pm on 21 April, 2025: User prefers the "compact literary classics" lane, enjoys works similar to those by Chinua Achebe and Buchi Emecheta, and wants books roughly 180–300 pages long.*

The answer model therefore sees an agent-organized input: persistent workspace instructions, plugin-recalled context injected by OpenClaw, and the current query as the user turn.

Difference. The memory backend can be the same, and the target query is unchanged, but the context organization differs. Standalone Mem0 exposes retrieval output as a flat benchmark `# CONTEXT` list inside the answer prompt. Mem0 + OpenClaw exposes plugin recall as runtime-injected agent context under OpenClaw workspace rules. This changes where instructions live, how retrieved memories are packaged, and how the final answer model encounters the same evidence.

Figure 31: Concrete context-organization example for a standalone Mem0 run and a Mem0 + OpenClaw run on the same target query.

LLM-as-judge prompt summary

System role. The judge acts as a benchmark answer judge and decides whether the generated answer should be labeled `CORRECT` or `WRONG`. It uses accepted correct answers as positive references, known incorrect answers as negative references, and relation metadata as auxiliary context.

Main grading rules.

- Mark `CORRECT` when the generated answer matches the meaning of an accepted correct answer, including paraphrases or complete longer explanations.
- Mark `WRONG` when the answer agrees with a known incorrect answer, contradicts the accepted answer set, misses the key point, or makes an unsupported overconfident choice.
- Use facts, case description, relation type, relation subtype, topic, source, and additional guidance to resolve grading ambiguity.
- For contradictory-relation and nuanced-relation cases, do not reward answers that remove the required distinction or conflict without evidence.

Required output. The prompt requires JSON with exactly two keys, `reason` and `label`.

Figure 32: Prompt summary for binary LLM-as-judge answer evaluation.

Soft Answer Prompt

Role. You are a helpful personal assistant. You are very good at distinguishing detailed conflicts and relationships in memory, then using those memories to answer questions or complete tasks.

 # CONTEXT
`{context}`

 # INSTRUCTIONS

- Answer the question or complete the user's requested task based on the provided context.
- First identify the information and details in the context that are useful for answering the question.
- If the useful information contains time-based updates, use the time mentioned by the user to decide which information applies.
- Some information may apply only in different situations, such as different ways of speaking in different professional roles.
- Some questions require using all useful information to provide a complete answer. In those cases, consider all relevant information when answering or completing the task.
- Do not treat the order of information in the context, or the chronological order of session timestamps, as proof that one piece of information is an update. The user may disclose older information in a later conversation, so do not trust only the latest session or the latest context item.
- When multiple pieces of information could each answer the question or help complete the task, but they would lead to different answers or outcomes and there is no clear way to decide, clarify the conflict.
- When several unconditioned pieces of information would lead to different answers or task outcomes, do not choose one directly; ask for clarification.

Question: `{question}`

Answer:

Figure 33: Soft answer prompt used for answer generation. The box preserves the main structure and rules of the prompt while omitting non-essential formatting lines.

Strong Answer Prompt

Role. You are a helpful personal assistant. You are very good at distinguishing detailed conflicts and relationships in recalled memories, then using those memories to answer questions or complete tasks.

 # CONTEXT

The information below provides only the context relevant to the current question. It may contain stable preferences, temporary states, factual claims, summaries, or contradictions.

`{context}`

 # INSTRUCTIONS

1. Use only the provided context. Do not use outside knowledge, common knowledge, or your own guess to resolve conflicts.
2. Your first priority is evidence fidelity: detect true unresolved conflicts, but do not over-detect conflicts from compatible evidence.
3. Identify the exact target needed by the question: preference, choice, attribute, state, factual answer, origin, date, name, or category.
4. Treat a conflict as mutually exclusive claims about the same target; different interests, different sources, background facts, or multiple constraints are not conflicts unless they directly support incompatible answers to the question target.
5. Search the context for evidence that supports one answer and evidence that supports a different or opposing answer for the same target, including semantic opposites with different wording.
6. Treat preference conflicts, incompatible choice evidence, factual answer conflicts, and conflicts between user statements and summaries as unresolved unless the context explicitly resolves them with a time frame, context, correction, current-state update, condition, or exception.
7. Do not assume the newest statement overrides older evidence unless the context explicitly says it is an update, correction, or current state, and do not invent a compromise, exception, or hierarchy to make conflicting evidence fit together.
8. If unresolved conflict affects a recommendation, reservation, purchase, registration, list, ranking, yes/no answer, or any other decisive answer, do not choose a side and do not choose a safer alternative. Start with `Unclear — needs clarification first.` Then briefly state both conflicting sides.
9. For factual questions with two different final answers for the same target, say the remembered answers conflict rather than selecting the answer that seems more correct from world knowledge. If the question is explicitly time-anchored or context-anchored and that anchor clearly selects one side, answer that side directly.
10. Treat evidence as compatible, and answer directly, when memories can all be true, one memory gives background while another answers the question, one memory adds a satisfiable constraint, different phrasings support the same answer, or the question requires combining multiple facts.
11. For multi-part or choice questions, combine compatible facts across the context and follow the option that best matches the question target without calling unrelated background interests a conflict.
12. Keep the answer concise: one or two sentences, without step-by-step reasoning.

Question: `{question}`

Answer:

Figure 34: Strong answer prompt used for answer generation. The box preserves the main structure and core conflict-handling rules of the prompt while omitting long output-pattern examples.

D.1.5 Oracle and Perfect-retrieval Protocols

The evidence settings differ only in what evidence reaches answer generation:

- **Oracle Setting.** The answer model receives the raw annotated target sessions $\mathbf{H}_\tau$ in chronological order, bypassing both memory formation and retrieval.
- **Perfect Retrieval Setting.** The system first writes the full history into memory. Retrieval is then replaced by provenance-guided read-back of the stored memory units written from $\mathbf{H}_\tau$, denoted $m_\tau = \mathcal{M}(\mathbf{H}_\tau)$.
- **Default Setting.** The target query is issued normally, and the system's own retrieval or recall mechanism determines the answer evidence.

Thus, the Oracle-to-Perfect gap measures whether answer-usable information survives memory formation, while the Perfect-to-Default gap measures whether preserved information is exposed by the default retrieval path.

D.2 Answer Generation Configuration

Answer generation is calibrated under oracle evidence before comparing memory systems. The calibration setting is:

- The calibration uses one complete user-history split with 141 evaluation queries.
- Each answer model receives the same raw target sessions $\mathbf{H}_\tau$ and target query, but no stored memory and no retrieved memory context.
- The same judge scores each answer against A^+, A^-, and relation metadata, so differences in Table 2 reflect answer-model and prompt behavior rather than storage or retrieval.
- The calibration compares `gpt-4o-mini`, `gpt-oss-120b`, and `gpt-5.4` under the soft and strong prompts. The strong prompt is used in the main evaluation because it reduces unsupported conflict resolution, especially for contradictory-relation cases.

D.3 Statistical Analysis of Main Comparisons

We compute uncertainty and significance tests as post-processing over the final question-level binary correctness labels. Overall confidence intervals use 10,000 nonparametric bootstrap resamples stratified by relation type. Relation- and subtype-level intervals use bootstrap resampling within the corresponding subset. For the main paired claim reported here, we compare Oracle Evidence against the best non-oracle system under each answer model using two-sided exact McNemar tests over aligned question IDs, with Holm correction across the two answer-model families. The best non-oracle system is selected by the overall point estimate in Table 1: Mem0 + OpenClaw for `gpt-5.4` and Mem0 for `gpt-oss-120b`. We align Oracle and the selected non-oracle run by `question_id`; each evaluation instance contributes one binary pair. McNemar tests use the discordant counts, and the reported Δ confidence interval is estimated with paired bootstrap resampling over the aligned questions. Table 13 reports the overall confidence intervals, and Table 14 reports the paired Oracle comparisons.

Method	Accuracy	95% CI
Base Model: GPT-5.4		
MemoBase	32.1%	[29.8%, 34.4%]
MIRIX	46.5%	[44.2%, 48.8%]
MemOS	56.8%	[54.5%, 59.1%]
Mem0	69.0%	[66.8%, 71.2%]
EverMemOS	68.1%	[66.0%, 70.2%]
A-Mem	70.0%	[67.8%, 72.1%]
MetaClaw	20.3%	[18.3%, 22.3%]
OpenClaw	62.5%	[60.4%, 64.6%]
MemOS + OpenClaw	56.5%	[54.3%, 58.6%]
Mem0 + OpenClaw	71.3%	[69.2%, 73.4%]
EverMemOS + OpenClaw	69.1%	[67.0%, 71.1%]
Oracle	85.4%	[83.7%, 87.1%]
Base Model: GPT-OSS-120B		
MemoBase	25.3%	[23.2%, 27.5%]
MIRIX	39.8%	[37.6%, 42.2%]
MemOS	52.2%	[49.9%, 54.5%]
Mem0	64.9%	[62.7%, 67.2%]
EverMemOS	60.4%	[58.2%, 62.7%]
A-Mem	63.9%	[61.7%, 66.2%]
MetaClaw	15.6%	[13.8%, 17.4%]
OpenClaw	52.3%	[50.1%, 54.6%]
MemOS + OpenClaw	47.8%	[45.5%, 50.1%]
Mem0 + OpenClaw	62.9%	[60.6%, 65.2%]
EverMemOS + OpenClaw	59.8%	[57.6%, 62.0%]
Oracle	75.7%	[73.8%, 77.5%]

Table 13: Overall main-table accuracy with 95% bootstrap confidence intervals. Each interval is computed from aligned question-level binary correctness labels, with bootstrap resampling stratified by relation type.

D.4 Repeated-run Robustness

To assess run-to-run stability, we repeat the complete evaluation pipeline for all six standalone mem-

Answer model	Δ pp	95% CI	Holm p
gpt-5.4	+14.1	[+12.0, +16.4]	4.96×10^{-34}
gpt-oss-120b	+10.8	[+8.5, +13.0]	5.41×10^{-20}

Table 14: Oracle Evidence versus the best non-oracle system in the main results. Δ is Oracle minus the best non-oracle point estimate in percentage points. Confidence intervals are paired bootstrap intervals, and p-values are two-sided exact McNemar tests with Holm correction.

Measure	Result
Mean absolute score change	0.85 pp
Maximum absolute score change	1.59 pp
Instance-level agreement	91.43–95.87%
Cohen's κ	0.826–0.905
Paired bootstrap 95% CIs	All include zero
Holm-adjusted McNemar p-values	All 1.000
Spearman's ρ	1.000
Kendall's τ	1.000

Table 15: Repeated-run robustness across six standalone memory systems on two persona histories. The repeated runs cover the complete evaluation pipeline, including memory formation, retrieval, answer generation, and judging.

ory systems on two persona histories using a new model seed under otherwise unchanged evaluation settings. The rerun includes each system's native memory formation and retrieval, answer generation, and answer judging.

For each system, we compare the original and repeated runs using the absolute change in overall accuracy, instance-level agreement, Cohen's κ, a paired bootstrap confidence interval for the accuracy difference, and an exact McNemar test with Holm correction. Table 15 summarizes the resulting run-to-run variation across the six standalone systems.

Across the six systems, the mean absolute score change is 0.85 percentage points and the maximum change is 1.59 points. Instance-level agreement ranges from 91.43% to 95.87%, with Cohen's κ ranging from 0.826 to 0.905. All paired bootstrap confidence intervals include zero, and all Holm-adjusted exact McNemar p-values are 1.000. The system ordering is unchanged, with both Spearman's ρ and Kendall's τ equal to 1.000. These results indicate small run-to-run variation on the evaluated two-history subset without changing the system-level conclusions.

Answer model	Spearman's ρ	Kendall's τ
gpt-5.4	0.955	0.887
gpt-oss-120b	0.970	0.920

Table 16: Mean correlations between persona-level method rankings and the global rankings across the 10 histories.

Answer model	Gap (pp)	95% CI	Holm p
gpt-5.4	14.1	[12.4, 16.0]	0.00977
gpt-oss-120b	10.8	[8.5, 13.0]	0.00977

Table 17: History-clustered Oracle gaps over the strongest non-Oracle system. Both gaps are positive across all 10 histories.

D.5 History-level Reliability

We conduct two history-level analyses to examine whether the main conclusions are driven by individual persona histories and whether within-history dependencies affect the reported Oracle gaps.

Persona-level consistency. We analyze the main-experiment results separately across all 10 persona-specific histories for the 12 evaluated methods under both answer backbones. Across the resulting 24 system–backbone configurations, the persona-level standard deviation averages 3.2 percentage points and ranges from 1.9 to 4.6 points. The overall accuracy differs from the corresponding unweighted persona mean by at most 0.2 points.

We additionally compare the method ranking within each history with the global ranking. As shown in Table 16, the resulting correlations remain high under both answer models.

These results indicate that the broad system-level performance structure is consistent across histories and is not driven by a single persona.

History-clustered Oracle gaps. To account for dependencies among questions and semantic cases from the same history, we conduct a paired history-cluster bootstrap with 10,000 resamples, grouping all instances by complete history. We conduct paired significance testing by enumerating all 2^{10} whole-history label swaps and apply Holm correction across five claim-aligned comparisons within each answer-model backbone.

Table 17 reports the history-clustered gaps between Oracle and the strongest non-Oracle system under each answer model.

The Oracle advantage remains positive across all 10 histories and significant after history clustering

and multiple-comparison correction, supporting the broad conclusion that current systems remain substantially below Oracle evidence access.

D.6 Answer-model and Judge Robustness

Answer-model robustness. To examine whether the system-level results depend on a particular answer model, we repeat the main evaluation protocol for all six standalone memory systems and Oracle on 315 questions from two persona-specific histories. In addition to `gpt-5.4` and `gpt-oss-120b`, we evaluate `Kimi-K2.5` and `Qwen3.7-Plus` as two additional non-OpenAI answer models. Table 18 reports the complete relation-wise results under the four answer models.

Across the four answer models, pairwise system-ranking correlations range from Spearman's $\rho = 0.899$ to 1.000, with a mean of 0.945. The same three standalone systems consistently form the leading group, while Oracle remains above the strongest non-Oracle system by 9.5–13.3 percentage points. Together, these results support the consistency of the broad system-level findings across the evaluated answer models.

Judge robustness. We further examine whether answer evaluation depends on the selected LLM judge by re-annotating 282 Oracle responses generated for a 141-question persona-specific history under two answer models. We use `Gemini 3.1`, `GLM-5.1`, `Qwen3.7-Plus`, and `Kimi-K2.5` as four independent judges. The four judges unanimously agree on 92.20% of the labels, with Fleiss' $\kappa = 0.848$ and pairwise agreement ranging from 93.97% to 98.58%. These results support the robustness of answer evaluation across the four evaluated judges.

D.7 Main Experiment Case Studies

Figures 35 and 36 break representative main-experiment cases into the underlying facts, relation type, baseline outputs, and judge decisions, without reproducing the full conversation sessions.

D.8 Perfect-retrieval Detailed Results

Table 19 reports the GPT-5.4 perfect-retrieval setting with the same relation and subtype columns as the main results table. In this setting, answer generation receives the stored memory units linked to the target evidence sessions, bypassing each system's default query-time retrieval path.

D.9 Representative Answer Examples

Figures 37 and 38 give representative correct and incorrect answer examples from baseline evaluation results. Each figure includes both a user-related example and a non-user example, and reports the original case, facts, reference answer, generated answer, and judge decision.

E Use of AI Assistants

AI assistance was used in three limited roles. First, LLMs supported benchmark construction by generating and filtering semantic variants, interaction sessions, and evaluation instances; the full construction procedure is documented in Appendix C. Second, our evaluation protocol uses an LLM-based automatic judge, with human-agreement validation reported in Appendix D.1.3. Third, general-purpose AI writing tools were used to improve wording and readability. They were not used to originate the research questions, choose the experimental design, produce the reported results, or draw the paper's conclusions. The authors reviewed and approved all benchmark design choices, analyses, results, and manuscript text.

Case 1: Complementary Any-one – coffee profile

- **Facts and target.** Lars drinks black coffee regularly, and when he has coffee he usually takes it without milk or sugar. The target asks for three compact staff-profile fields: coffee selection, milk/cream, and sweetener.
- **Relation type.** Complementary / Any-one: either fact is sufficient to identify the same plain-coffee preference.
- **Accepted answer.** Black brewed coffee; no milk or cream; no sugar.
- **Baseline outputs.** Mem0 (GPT-5.4) answers "one large black mug of coffee," with no milk/cream and no sugar, and is judged correct. Oracle (GPT-5.4) answers straight filter coffee with no milk/cream and no sugar, also correct. MemOS (GPT-5.4) instead writes "preferred coffee format for alertness," "milk-light," and no specific sweetener note, so it is judged wrong.
- **Judge decision.** The judge accepts semantically equivalent plain-black-coffee answers and rejects outputs that miss or contradict the no-milk/no-sugar fields.

Case 2: Nuanced Contextual – short-story writing

- **Facts and target.** Malik writes politically sharp op-eds for independent magazines, but he deliberately keeps short fiction unrelated to politics as a separate creative space. The target asks how to get a short story moving this week.
- **Relation type.** Nuanced / Contextual: the political-writing fact is related but belongs to a different writing context, while the fiction fact controls the current task.
- **Accepted answer.** Keep the story away from politics; focus on character, image, scene, and concrete detail rather than slogans or argument.
- **Baseline outputs.** Oracle (GPT-5.4) explicitly says the plan should be personal and concrete, with no debate scenes or slogan energy, and is correct. MemOS (GPT-5.4) says to use a low-pressure, observation-based routine that avoids slogans or policy talk, and is also correct.
- **Judge decision.** The judge rewards selecting the fiction-specific memory rather than applying Malik's op-ed persona to the story-writing task.

Figure 35: Representative main-experiment cases for complementary and nuanced relations, showing the facts, relation type, selected baseline outputs, and judge decisions.

Case 3: Contradictory – nature-documentary preference

- **Facts and target.** Julian is remembered both as liking nature documentaries for unwinding and as finding nature documentaries boring and never choosing them. The target asks for a hotel media-preference card.
- **Relation type.** Contradictory: the target-relevant facts make opposite claims under the same media-preference target.
- **Accepted answer.** Leave the nature-documentary setting unconfirmed, hold the card, and ask whether nature documentaries should be enabled or excluded.
- **Baseline outputs.** Oracle (GPT-5.4) states that the context points both ways, sets the field to needs clarification, and is judged correct. MemOS (GPT-5.4) resolves the conflict into "leave off recommendation list" and "send now," so it is wrong. Mem0 (GPT-5.4) says there are no remembered media preferences and repeats formatting instructions, so it is also wrong.
- **Judge decision.** The judge marks a case correct only when the answer preserves the conflict and asks for confirmation instead of choosing one side or claiming no evidence exists.

Case 4: Complementary Any-one – book pickup list

- **Facts and target.** Amara enjoys African literature classics and often gravitates toward classic novels by African authors. The target asks for two books from an already-approved hold shelf.
- **Relation type.** Complementary / Any-one: multiple facts point to the same selection criterion, so any sufficient remembered variant can support the answer.
- **Accepted answer.** *So Long a Letter* and *The Concubine*, because they are African literary classics and fit the shelf.
- **Baseline outputs.** MemOS (GPT-5.4) selects *So Long a Letter* and *The Concubine*, and is judged correct. Oracle (GPT-5.4) selects the same two titles, also correct. Mem0 (GPT-5.4) selects *So Long a Letter* but replaces *The Concubine* with *A Month in the Country*, so it is wrong.
- **Judge decision.** The judge requires both the right titles and the right memory relation: the titles must be selected because they match the African-literary-classics preference, not merely because they are short or easy weekend reads.

Figure 36: Representative main-experiment cases for contradictory and relation-critical complementary examples, showing the facts, relation type, selected baseline outputs, and judge decisions.

Representative correct-answer examples

Example 1: User-related resource arrangement.

System setting. MemOS with GPT-5.4.

Original case. Amara uses different design styles depending on whether she is shaping her home environment or creating materials and spaces for kids' science activities.

Original facts. (1) At home, Amara prefers minimalist, Scandinavian-inspired design with clean lines, light wood tones, and an uncluttered feel. (2) For children's STEM workshops or toy-car race setups, she uses bold colors, interactive visual cues, and playful layouts.

Golden answer. Signal Sprint Kit — it matches bold colors, interactive cues, and small activity islands for an energetic kids setup.

Observed generated answer. Signal Sprint Kit — it best matches the remembered brief: bright lane colors, big visual cues, and separate activity islands support intuitive child navigation and fast engagement without relying on dense text.

Judge label/reason. CORRECT: the generated answer correctly selects the Signal Sprint Kit and provides a reason that aligns with the accepted correct answers regarding its suitability for a children's activity.

Example 2: Non-user multi-evidence question.

System setting. Mem0 with GPT-5.4.

Original case. What are the names of 6 Metropolitan cities in Korea and their respective symbol flowers?

Original facts. (1) Busan's symbol flower is Camellia. (2) Incheon's symbol flower is Rose. (3) Ulsan's symbol flower is Pear flower.

Golden answer. Busan's symbol flower is Camellia, Incheon's is Rose, and Ulsan's is Pear flower.

Observed generated answer. Busan — Camellia; Incheon — rose; Ulsan — Pear flower.

Judge label/reason. CORRECT: the generated answer correctly identifies the symbol flowers for Busan, Incheon, and Ulsan, matching the accepted correct answers.

Figure 37: Representative correct-answer examples from baseline SubtleMemory evaluation results. The examples cover both user-related and non-user sources.

Representative incorrect-answer examples

Example 1: User-related language archive card.

System setting. MemOS with GPT-5.4.

Original case. Amara enjoys learning languages casually, but the language she has focused on in her spare time has shifted over the years.

Original facts. (1) When she first moved to the U.S. for graduate school, Amara casually practiced French to keep using the basics she already knew. (2) More recently, Amara has been putting that casual language-learning energy into strengthening her conversational Igbo to feel more connected to her family and heritage.

Golden answer. Time period: first move to the U.S. for grad school; casual language practice: French basics; why I kept it up: to keep using what I already knew; intensity level: casual.

Observed generated answer. Time period: first move to the U.S. for graduate school; casual language practice: random podcasts, science-themed narratives; why I kept it up: made the apartment feel less quiet, more comfortable; intensity level: light, casual. . . .

Judge label/reason. WRONG: the generated answer includes incorrect information for the language practiced and the reason for keeping it up, failing to mention French basics and maintaining prior knowledge.

Example 2: Non-user contradictory question.

System setting. Mem0 with GPT-5.4.

Original case. What type of book is *The Fault in Our Stars*?

Original facts. (1) What type of book is *The Fault in Our Stars* when describing intended age group? Young adult. (2) What type of book is *The Fault in Our Stars* when describing general content? Realistic fiction.

Golden answer. The remembered sessions conflict: one says *The Fault in Our Stars* is young adult, while another calls it realistic fiction, so the type remains unresolved without clarification.

Observed generated answer. The context supports calling *The Fault in Our Stars* a young adult novel; on the user's classroom display card, its genre line is listed as realistic fiction.

Judge label/reason. WRONG: the generated answer invents details about a classroom display card to resolve the conflict, rather than acknowledging the unresolved contradiction as required by the accepted answers.

Figure 38: Representative incorrect-answer examples from SubtleMemory evaluation results. The examples cover both user-related and non-user sources.

Method	Complementary			Nuanced			Contradictory	All
	Multi-evidence	Any-one	Overall	Temporal	Contextual	Overall		
Answer Model: GPT-5.4								
MemoBase	20.9%	37.9%	27.8%	33.0%	43.6%	38.0%	7.8%	28.3%
MIRIX	51.2%	48.3%	50.0%	50.0%	70.5%	59.6%	10.4%	45.4%
MemOS	48.8%	**55.2%**	51.4%	72.7%	79.5%	75.9%	20.8%	56.8%
Mem0	**62.8%**	48.3%	**56.9%**	**81.8%**	**91.0%**	**86.1%**	42.9%	**68.9%**
EverMemOS	58.1%	51.7%	55.6%	80.7%	89.7%	84.9%	35.1%	66.0%
A-Mem	58.1%	41.4%	51.4%	80.7%	88.5%	84.3%	**48.1%**	67.9%
Oracle	69.8%	79.3%	73.6%	95.5%	93.6%	94.6%	63.6%	82.2%
Answer Model: GPT-OSS-120B								
MemoBase	16.3%	44.8%	27.8%	26.1%	35.9%	30.7%	5.2%	23.8%
MIRIX	37.2%	48.3%	41.7%	42.0%	64.1%	52.4%	5.2%	38.4%
MemOS	51.2%	**55.2%**	52.8%	69.3%	73.1%	71.1%	19.5%	54.3%
Mem0	**58.1%**	48.3%	**54.2%**	**83.0%**	82.1%	**82.5%**	**40.3%**	**65.7%**
EverMemOS	53.5%	51.7%	52.8%	75.0%	**85.9%**	80.1%	20.8%	59.4%
A-Mem	**58.1%**	37.9%	50.0%	79.5%	83.3%	81.3%	39.0%	63.8%
Oracle	79.1%	82.8%	80.6%	92.0%	92.3%	92.2%	39.0%	76.5%
Answer Model: Kimi-K2.5								
MemoBase	18.6%	37.9%	26.4%	26.1%	43.6%	34.3%	5.2%	25.4%
MIRIX	41.9%	34.5%	38.9%	45.5%	66.7%	55.4%	6.5%	39.7%
MemOS	44.2%	48.3%	45.8%	51.1%	73.1%	61.4%	20.8%	47.9%
Mem0	**67.4%**	48.3%	**59.7%**	83.0%	84.6%	83.7%	29.9%	65.1%
EverMemOS	55.8%	44.8%	51.4%	80.7%	**89.7%**	**84.9%**	16.9%	60.6%
A-Mem	60.5%	**51.7%**	56.9%	**85.2%**	84.6%	**84.9%**	**40.3%**	**67.6%**
Oracle	79.1%	82.8%	80.6%	93.2%	93.6%	93.4%	49.4%	79.7%
Answer Model: Qwen3.7-Plus								
MemoBase	9.3%	44.8%	23.6%	28.4%	41.0%	34.3%	6.5%	25.1%
MIRIX	44.2%	55.2%	48.6%	50.0%	64.1%	56.6%	2.6%	41.6%
MemOS	48.8%	51.7%	50.0%	63.6%	75.6%	69.3%	20.8%	53.0%
Mem0	**65.1%**	44.8%	56.9%	85.2%	87.2%	86.1%	36.4%	67.3%
EverMemOS	62.8%	**58.6%**	**61.1%**	**86.4%**	**91.0%**	**88.6%**	27.3%	67.3%
A-Mem	58.1%	**58.6%**	58.3%	83.0%	87.2%	84.9%	**46.8%**	**69.5%**
Oracle	81.4%	79.3%	80.6%	96.6%	97.4%	97.0%	39.0%	79.0%

Table 18: Answer-model robustness on 315 questions from two persona-specific histories. Results cover six standalone memory systems and Oracle under four answer models and are reported by relation type. Best and second-best non-Oracle point estimates within each answer-model block are bolded and underlined, respectively.

Method	Complementary			Nuanced			Contradictory	All
	Multi-evidence	**Any-one**	**Overall**	**Temporal**	**Contextual**	**Overall**		
Base Model: GPT-5.4; Setting: Perfect Retrieval								
Mem0	75.4%	82.7%	78.4%	88.9%	91.5%	90.2%	51.7%	77.9%
MemOS	64.0%	81.3%	71.2%	79.3%	85.6%	82.4%	31.6%	67.1%
EverMemOS	76.8%	80.7%	78.4%	89.1%	92.8%	90.9%	44.3%	76.4%
MIRIX	66.8%	74.7%	70.1%	68.7%	84.5%	76.5%	15.9%	60.0%
A-Mem	77.7%	86.0%	81.2%	92.9%	93.0%	93.0%	**61.5%**	**82.4%**
MemoBase	34.6%	53.3%	42.4%	37.4%	47.9%	42.6%	18.8%	36.7%
OpenClaw	79.1%	86.0%	82.0%	**95.2%**	**95.4%**	**95.3%**	51.2%	81.2%
MetaClaw	18.5%	70.7%	40.2%	47.2%	52.3%	49.7%	5.3%	36.5%
Mem0 + OpenClaw	79.6%	**88.0%**	**83.1%**	90.2%	92.3%	91.2%	19.6%	71.6%
MemOS + OpenClaw	69.2%	85.3%	75.9%	76.5%	86.6%	81.5%	16.4%	64.1%
EverMemOS + OpenClaw	**80.1%**	83.3%	81.4%	89.4%	91.0%	90.2%	45.1%	76.9%

Table 19: GPT-5.4 perfect-retrieval results on SubtleMemory. Results are reported with the same relation and subtype columns as the main results table. Best and second-best values among non-oracle baselines are bolded and underlined, respectively.