BEYOND MEMORIZATION: BENCHMARKING MEMORY UTILIZATION IN CONVERSATIONAL LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) agents increasingly rely on persistent cross-session memory to support long-horizon and personalized tasks. To understand how memory is actually needed in real-world use, we annotated WildChat and identified 21,382 memory-dependent queries, of which 83.7% do not ask about memory content directly but invoke it while solving downstream tasks. Existing memory benchmarks almost exclusively evaluate the former: whether memory can be accurately recalled, leaving the latter untested. Drawing on Bloom's taxonomy, we identify three higher-order memory abilities beyond basic recall: Comprehension of memory revealed in dialogue, Apply of memory to downstream tasks, and Evaluation of whether memory should be invoked or used to correct erroneous queries. We introduce **MUSE-Bench**, which operationalizes these abilities into six utilization scenarios under a need-first construction pipeline. Built upon 100 carefully constructed user profiles spanning 12 life domains, **MUSE-Bench** contains 3,840 queries across six evaluation tasks. Notably, the Evaluation level introduces two opposing failure modes: Interference Memory, where irrelevant or misleading memory must be suppressed; and Counterfactual Query, where memory must be invoked to correct factual errors in the query, targeting the deployment risks of memory overuse and sycophancy. We evaluate 14 frontier LLMs and 20 memory baselines on **MUSE-Bench**, and identify two key findings: i) Frontier LLMs perform reasonably at Comprehension but show substantial gaps at Apply and Evaluation level; ii) Existing memory techniques offer limited improvement in memory utilization over conventional RAG methods. These results suggest that *the next frontier of agent memory is not memorizing more, but using more wisely*[1].

1 INTRODUCTION

Large Language Models (LLMs) agents are evolving from single-turn tools into long-horizon, cross-session personal assistants capable of supporting users across days, weeks, or even months of interaction (Yao et al., 2022; Park et al., 2023b; Xi et al., 2025). Underlying this transformation is persistent memory, the mechanism that allows an agent to accumulate user information, prior interaction history, and task context across sessions (Packer et al., 2023; Zhong et al., 2024; Wang et al., 2024a). As memory-augmented agents move from research prototypes into real deployment, the reliability of their memory capabilities has become critical to user experience and trust. Therefore, systematically evaluating what agents can actually do with memory is an unavoidable question.

To address this evaluation gap, several memory benchmarks have been proposed (He et al., 2026; Maharana et al., 2024; Ai et al., 2025). For instance, LongMemEval (Wu et al., 2025) evaluates multi-session memory recall across long conversational histories. Despite their different emphases, most of them follow a common design principle: *Queries explicitly target the stored information* (e.g., " What did I tell you about my commute time? "). Under this design, the core challenge for the agent is to accurately retrieve the targeted information from history.

[1]Both the evaluation code and the dataset are anonymously released. Code: https://anonymous.4open.science/r/MUSE-Bench-FC48/README.md; Dataset: https://huggingface.co/datasets/anonymous111111111/MUSE-Bench

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

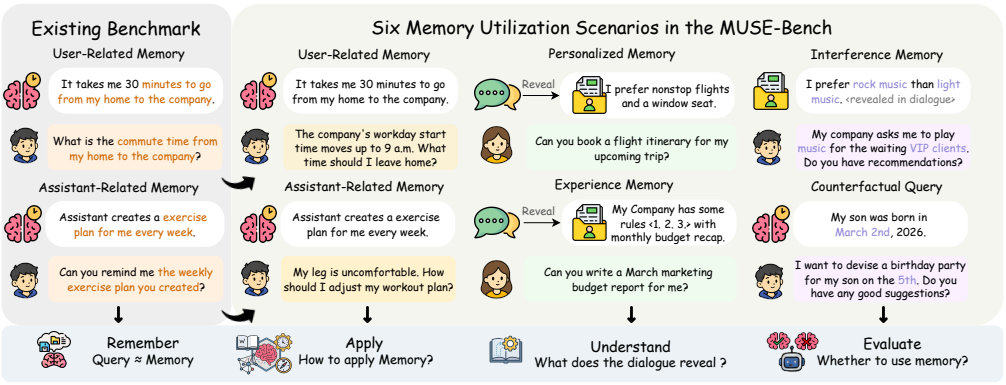


Figure 1: From Remember to Utilization: **MUSE-Bench** extends evaluation across four cognitive levels. Inspired by Bloom's taxonomy, while existing benchmarks (Left) focus on Remember, **MUSE-Bench** (Right) evaluates three higher-order abilities: Apply, Understand, and Evaluate.

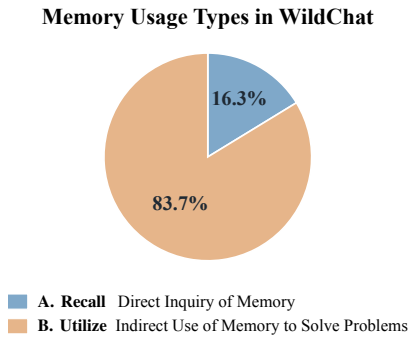


Figure 2: Memory Usage Types in WildChat, a real in-the-wild user interactions dataset with ChatGPT. More experimental details can be found in Appendix A.

However, the queries that deployed agents actually encounter differ systematically from what these benchmarks measure. Two paradigms can be distinguished. Under the *"memory-as-answer"* paradigm, the stored information is itself the target of the query, and accurate retrieval is the core challenge; this is the setting most existing benchmarks instantiate (Maharana et al., 2024; Wu et al., 2025). Under the *"memory-as-premise"* paradigm, memory instead supports a downstream task, and the agent must decide whether, when, and how to use it. Consider a user who once told his assistant about his commute time. The next day, he asks: "My company moved the workday start to 9 a.m. What's the latest time I should leave home?" Answering correctly requires the stored commute duration, yet the query gives no cue pointing to it: it mentions neither commuting nor the earlier conversation, and shares no surface overlap with the stored fact. Retrieval accuracy alone does not settle the matter. The agent must first recognize that the question depends on memory at all, and a relevance-keyed retriever provides no signal for that judgment. This is precisely the decision that memory-as-answer evaluation leaves untested. Among 21,382 memory-dependent queries we annotate in WildChat (Zhao et al., 2024), 83.7% take this indirect form (Figure 2), which reflects the latent demand of the user. While existing evaluations offer little signal about how agents actually utilize memory in deployment.

To operationalize the "*memory-as-premise*" paradigm, we draw on Bloom's taxonomy from cognitive science (Anderson & Krathwohl, 2001), which organizes cognitive abilities into four levels: Remember, Understand, Apply, and Evaluate. Mapped to memory in LLM agents, Remember corresponds to the recall scope of existing benchmarks. Building on this foundation, there are three higher-order capabilities: Understand, inferring users' personalized preferences or experiential requirements from dialogue; Apply, applying recalled or inferred memory to novel downstream tasks; and Evaluate, judging whether memory should be invoked, including the opposing capabilities of suppressing irrelevant memory and using memory to correct erroneous queries. Among recent works, PersonaMem and PrefEval (Zhao et al., 2025a; Jiang et al., 2025a) test whether agents can infer user preferences from dialogue history for recommendation. However, they cover only the personalization slice of Understand, and Apply and Evaluate remain systematically unevaluated.

Building on this, we propose a **M**emory **U**tilization **E**valuation **Bench**mark, termed **MUSE-Bench**, which operationalizes these abilities into six utilization scenarios under a need-first construction pipeline. As is shown in Figure 1, **MUSE-Bench** comprises 3,840 queries across six real-world scenarios, grounded in 100 carefully constructed user profiles spanning 12 life domains. Notably, the Evaluate level introduces two opposing failure modes: Interference Memory, where irrelevant memory must be suppressed; and Counterfactual Query, where memory must correct factually erroneous queries, directly targeting the deployment risks of memory misuse and sycophancy. We evaluate 14 frontier LLMs and 20 memory baselines on **MUSE-Bench**, and find that: i) Frontier LLMs perform reasonably

at the Understand level but exhibit substantial gaps at the Apply and Evaluate levels; ii) Existing training-free memory techniques offer limited improvement in memory utilization over conventional RAG methods. Our contributions are as follows:

- We propose a Bloom-grounded memory competence framework that operationalizes "memory utilization" into three independently evaluable abilities.
- We introduce **MUSE-Bench**, a benchmark of 3,840 multiple-choice queries grounded in 100 personas and 12 life domains, covering six utilization scenarios that instantiate the three abilities, released at 128K and 1M context scales with an identical query set.
- We conduct a systematic evaluation of 14 frontier LLMs and 20 memory baselines, exposing the limited memory utilization ability of both frontier LLMs and current memory methods.

2 BENCHMARK

2.1 PROBLEM FORMULATION

MUSE-Bench evaluation is defined over a 4-tuple (P, S, q, t_q), where P is a *user persona* comprising static demographics and *dynamic* characteristics such as evolving preferences, habits, and social experience; $S = [(t_1, S_1), \ldots, (t_N, S_N)]$ is a temporally ordered sequence of N multi-turn chat sessions with timestamps, each covering one of 12 life domains and potentially carrying, exposing, updating, or contradicting pieces of the user's evolving memory; and q is an *in-situ* first-person query issued at date $t_q > t_N$. Given S, the model must produce the response most consistent with the user's memory, formulated as a multiple-choice selection that enables deterministic accuracy scoring while requiring the model to recall, integrate, update, and disambiguate user memory across long histories.

2.2 SIX MEMORY UTILIZATION SCENARIOS

A reliable personalized assistant must do more than memorize facts: it must decide *what* to learn from the dialogue, *how* to apply it, and *whether* to use it when conflicts arise. To capture this, **MUSE-Bench** is organized along three high-level memory abilities: Understand, Apply, and Evaluate, which together instantiate six memory utilization scenarios. The taxonomy is illustrated in Figure 1.

(I) Understand. This level evaluates whether an assistant has internalized what it has built up about a user over long-term interaction. Crucially, such understanding is not confined to personalization: existing benchmarks are built around modeling user preferences, with the aim of producing a personalized response. Yet a long-term collaborator must grasp not only what the user prefers, but also how their everyday tasks are actually carried out. When the same kind of task recurs and the same requirement is imposed on it each time, that regularity is equally revealing. We accordingly decompose this level into *personalized memory* and *experience memory*.

- **Personalized Memory.** Users carry persistent traits into their interactions, spanning stable preferences, habitual routines, and relational facts about people in their lives. These are rarely stated outright, but revealed through the choices a user makes across sessions. A user would never ask what kind of seat they prefer, since that much is already settled for them; but they may well ask the assistant to book a flight, and the query omits the very trait it depends on. The model must therefore infer it from behavioral signals scattered across the history.
- **Experience Memory.** Through repeated task-oriented interactions, the assistant accumulates the constraints a user works under: structural templates, content requirements, and quality criteria that define how a task should be completed. Such constraints are rarely preferences, but demands user is under, set by an employer or an institutional rule. When a similar task arrives, the model must synthesize these cues into reusable procedural knowledge and apply the constraint unprompted.

(II) Apply. Apply shifts the question from what an assistant has come to understand of a user to whether that understanding is summoned when a task calls for it. This level evaluates whether, when a request depends on remembered content as a necessary step toward its answer, the assistant recognizes as much of its own accord and then recalls and applies that content correctly. Existing benchmarks largely put their questions to memory directly, asking how long the user's commute is. Such questions show that the content has been recalled, but not that the assistant would draw on it unprompted. Real requests seldom take that form: a user whose check-in time moves up asks when to set out, and the commute has to be recalled without ever being mentioned. Recall and application are in this sense distinct capacities. The memory drawn upon may originate with the user or with the assistant, so we decompose this level into *user-related* and *assistant-related memory*.

Table 1: Comparison of **MUSE-Bench** with existing memory benchmarks. Under *Statistics*, *#Sess* stands for *Session*. Under the three ability groups, *Pers.* and *Exp.* denote *Personalized* and *Experience* memory; *User* and *Asst.* denote *User-Related* and *Assistant-Related* memory; *Int.* and *Ctf.* denote *Interference Memory* and *Counterfactual Query*. *Obj.* and *Subj.* stand for *Objective* and *Subjective*. All listed benchmarks cover the Remember level, which is therefore omitted from the table.

Benchmark	Data Statistics				Understand		Apply		Evaluate		Query	
	#Sess	#User	#Query	Avg Context	Pers.	Exp.	User	Asst.	Int.	Ctf.	Obj.	Subj.
MemoryBank (Zhong et al., 2024)	300	30	200	5K	✗	✗	✗	✗	✗	✗	✓	✗
PerLTQA (Du et al., 2024)	3K	32	8,593	15K	✗	✗	✗	✗	✗	✗	✓	✗
LoCoMo (Maharana et al., 2024)	272	20	1,986	20K	✗	✗	✓	✗	✗	✗	✓	✗
LongMemEval (Wu et al., 2025)	50K	NA	500	1.5M	✗	✗	✓	✗	✗	✗	✓	✗
PrefEval (Zhao et al., 2025a)	3K	NA	3,000	100K	✓	✗	✗	✗	✗	✗	✗	✓
PersonaBench (Tan et al., 2025)	1.4K	28	855	73K	✓	✗	✗	✗	✗	✗	✓	✓
PersonaMem (Jiang et al., 2025a)	1.8K	20	2,674	1M	✓	✗	✗	✗	✗	✗	✗	✓
RPEval (Feng et al., 2026)	953	300	300	1.8K	✗	✗	✗	✗	✓	✗	✗	✓
MUSE-Bench (Ours)	**75K**	**100**	**3,840**	**1M**	✓	✓	✓	✓	✓	✓	✓	✓

- **User-Related Memory.** The user has shared personal information across prior sessions, and some piece of it is a necessary input to resolving the current request. The memory is put to need in two ways: through quantitative or deductive reasoning over what the request newly supplies, or by pairing it with common knowledge anyone could contribute, such as a unit conversion or a standard formula. We vary difficulty along two dimensions: how many memories the answer depends on, since several may need to be combined; and whether temporal reasoning is required, where the answer turns on when things were recorded or how a situation has changed since.
- **Assistant-Related Memory.** What the assistant must draw on is not confined to what it has learned about the user; its own earlier outputs serve equally well as memory. Over the course of collaboration the assistant produces content of its own, such as a plan laid out in steps, which the user may later want adjusted or built upon. Such a kind of memory is designed to test whether the model can operationalize its own history.

(III) Evaluate. Evaluate turns from whether remembered content is brought to bear to whether it ought to be. The other two scenarios reward drawing on memory, yet this disposition is not always right: what is remembered may be genuine, current, and squarely on topic, and still not serve the case at hand. This level therefore evaluates whether an assistant weighs what it holds against the situation before it, rather than deferring by default either to memory or to the request. In the first case, the record is accurate and matches the topic the request raises, yet does not fit the context in which it is made. In the second, the user asserts something about their own circumstances that the record contradicts, testing whether the assistant accepts the premise or recognises the discrepancy. We accordingly decompose this level into the following two classes:

- **Interference Memory.** The history contains a preference or habit that is accurate and on topic, yet placed in a setting whose conventions do not align with it. The preference and the setting bear on the same dimension, so that applying what is remembered produces a concrete answer that is wrong for the case rather than merely beside the point. The model must detect the mismatch and refrain from applying the memory, rather than forcing a superficially relevant recall into its response.
- **Counterfactual Query.** The in-situ query contains a premise about the user's situation that contradicts what the history records. An assistant that takes the user at their word lands on an answer that is identifiably wrong, and does so without any sign that its own record says otherwise. What is measured is whether the mismatch is noticed and raised at all: the model must surface the inconsistency rather than passively accommodating the erroneous premise.

2.3 Benchmark Statistics

As is shown in Table 1, **MUSE-Bench** is built around 100 distinct personas spanning a wide range of demographics and occupations. The benchmark contains 3,840 multiple-choice queries in total, distributed across six fine-grained query types and 12 life domains, including *Food, Health, Work, Interpersonal, Shopping, Travel, Family, Learning, Leisure, Finance, Domestic*, *Technology*, so that no single topic dominates evaluation. To stress LLMs at different long-context regimes, we release **MUSE-Bench** at two scales that share identical query sets but differ in interaction-history volume: **MUSE-128K** provides 200 sessions per persona totaling approximately 128K tokens of

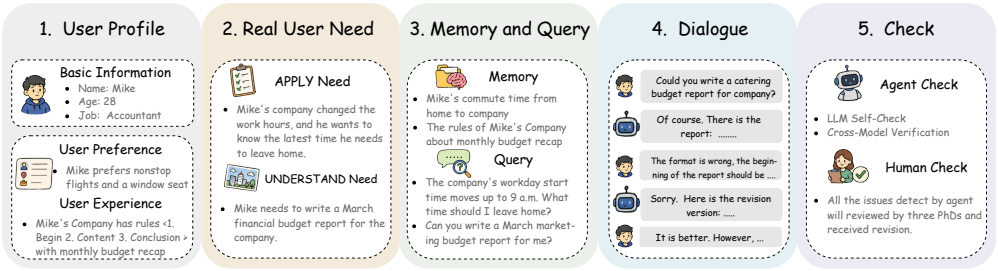


Figure 3: The Construction Pipeline of **MUSE-Bench**.

context, and **MUSE-1M** provides 750 sessions per persona totaling approximately 1M tokens of context. Each session is timestamped and topic-coherent, and sessions are interleaved in temporal order to simulate a long-running personal assistant deployment. This aligned-query design enables controlled comparison of how performance degrades as the memory horizon grows, isolating the effect of context length from question difficulty. The fully synthetic construction pipeline is readily extensible to additional personas, domains, or longer contexts without changing the evaluation protocol.

2.4 Benchmark Construction

Existing long-term memory benchmarks follow a fixed memory-first pipeline: a synthetic dialogue history is generated first, and queries are then derived from whatever that history contains. Since the memory is fixed before any query exists, every query must be answerable from it alone, and the space of admissible questions narrows to what the stored content can support on its own: asking back the memorized fact, or selecting a response consistent with a recorded preference. The recall orientation of these benchmarks thus follows from the construction order itself, rather than from a deliberate choice of scope. **MUSE-Bench** inverts the order under a **need-first, persona-grounded** principle. We begin from a rich user persona and identify situations in which that user would genuinely rely on an assistant's memory in the course of daily life; only then do we back-engineer the memories those situations require and generate the dialogue history that carries them. Because every memory originates from a need, memory is never the endpoint of the question but an input to it, and the evaluation stays on how it is actually put to use. The pipeline consists of five sequential stages, *Persona*, *Need*, *Memory & Query*, *Dialogue*, and *Quality Check*, as shown in Figure 3. We summarize each stage below and defer the full protocol to Appendix B.

User Profile. We build a *personal skeleton* for each user in three layers. The basic information layer is sampled by a controlled script to ensure balanced demographics and reproducibility. An LLM then enriches this skeleton with a user preference layer capturing stable likes, habits, and daily-life routines, and a user experience layer capturing profession-related procedural knowledge and long-running responsibilities. This skeleton anchors every subsequent generation step, so that one persona corresponds to *one coherent user lifetime* rather than disjoint synthetic fragments.

User Need. A *need* is the situation that brings a user to the assistant, described before any wording is chosen: what the user is trying to do, and why the assistant's memory bears on it. It is the basepoint of our construction: memories, queries, and distractors are all reverse-engineered from it. Needs are generated separately for each of the six scenarios, conditioned on all three profile layers so that the situation is one this specific persona would naturally find themselves in. For *Understand*, the user wants a routine task carried out, and doing it well depends on a stable trait or on a requirement their work is subject to, neither of which they would think to mention. For *Apply*, the user's circumstances have changed in a way that can only be worked out against something already on record, whether their own or the assistant's. For *Evaluate*, the user is in a situation where following memory would be the wrong move: either the occasion overrides a standing preference, or they have misremembered something the record settles. Because a flawed need propagates silently into every artifact derived from it, each instance passes a cross-model audit before any memory is generated, followed by manual inspection of a stratified sample. Per-scenario criteria and pass rates are given in Appendix B.2.

Memory and Query. For each accepted need, we derive the minimum set of *answer memories* required to produce the correct response, and additionally synthesize *scaffolding memories*: persona-consistent background facts that answer no query, but compose the texture of a real user's long-running life. A cross-consistency pass resolves contradictions across all memories before they enter the dialogue stage. Each need is then rendered into a first-person query, filtered to remove items

that leak the answer-bearing memory or that are consistent with multiple memories. We frame every item as a 5-way multiple-choice question rather than open-ended generation: at 1M-token context, LLM-as-judge calls scale with both item count and prompt length, while lexical metrics are unreliable for personalized responses, where correctness hinges on which user fact was used rather than surface wording. Options also let us probe specific failure modes. Each item's three distractors are drawn from four canonical modes, namely other-persona preference, outdated memory, premise-following, and generic default, while the fifth option is *I don't know*, which prevents models from guessing when they lack the relevant memory. Detailed definitions are provided in Appendix B.

Dialogue. Each memory is rendered into a topic-coherent multi-turn conversation between the user and the assistant. A substantial volume of filler sessions is injected to make retrieval nontrivial at test time. All sessions are timestamped along a multi-month timeline and interleaved across topics. The same memory and query set is re-woven at two filler densities to produce **`MUSE-128K`** and **`MUSE-1M`**, enabling controlled comparison across context budgets while holding the underlying questions fixed.

Quality Check. Every generated item passes a three-stage verification loop so that no single model is responsible for both generating and approving an item. First, the generation model (GPT-5.2) re-audits each item for internal consistency, answer leakage, and persona coherence. Second, an independent model (Gemini-3-flash) checks the surviving items to catch model-family-specific blind spots such as paraphrased leakage and temporal inconsistencies. Third, we validate the final benchmark with human evaluation: three trained annotators independently assess a stratified random sample of 500 items, and an item passes only when at least two approve. The overall pass rate is 96.4%, with Fleiss' $\kappa = 0.87$ across five criteria.

3 Experiment

3.1 Experimental Setup

Evaluation Protocol. Following Jiang et al. (2025a); Zhao et al. (2025a), every item is framed as a 5-way multiple-choice question: four candidate answers (A–D) and a fifth option (E: *I don't know*). Among A–D, one is the gold answer, and three are negative distractors drawn from canonical failure modes. Multiple-choice scoring is deterministic and reproducible, avoiding LLM-as-judge calls that scale with item count and prompt length at 1M-token context, as well as the unreliability of lexical metrics for personalized responses. Option E prevents models from guessing among plausible options when they lack the relevant memory. A model receives credit only for selecting the gold answer, which we report as accuracy (ACC). We additionally report the hallucination rate (HR), the share of items on which the model commits to a wrong option among A–D rather than abstaining via E.

Evaluated LLMs. We evaluate 14 LLMs spanning five model families: GPT-5.4 and GPT-5.2 (OpenAI); Claude Opus 4.6 and Claude Sonnet 4.6 (Anthropic); Gemini-3-Flash, Gemini-3.1-Flash-Lite, and Gemini-2.5-Flash (Google); Qwen3-Max, Qwen3.6-Plus, and Qwen3.5-Plus (Alibaba); GLM-5 and GLM-4.7 (Zhipu); Kimi-K2.5 (Moonshot) and MiniMax-M2.5 (MiniMax).

Baselines. We compare against 20 baselines from three categories. *Five context-utilization baselines*: Full-Context, NaiveRAG (Lewis et al., 2020), RAG-Fusion (Rackauckas, 2024), InstructRAG (Wei et al., 2025), and GenerativeAgent (Park et al., 2023c). *Eleven training-free memory methods*: MemGPT (Packer et al., 2023), MemoryBank (Zhong et al., 2024), LangMem[2], Mem0 (Chhikara et al., 2025), SimpleMem (Liu et al., 2026a), Secom (Pan et al., 2025), A-Mem (Xu et al., 2025b), HippoRAG2 (Gutiérrez et al., 2024; 2025), MemGAS (Xu et al., 2025a), MemoryOS (Kang et al., 2025), and LightMem (Fang et al., 2025). *Four trainable memory methods*: Mem-α (Wang et al., 2025b), MemGen (Zhang et al., 2025a), MemAgent (Yu et al., 2025), and MemCoE (Xu et al., 2026).

Implementation Details. The input to each model is the user's full interaction history, followed by the query. All models and baselines share the same prompt template. For RL-based methods, we use the first 20% of users for training and evaluate on the remaining 80%. For fairness, RAG-based methods are also evaluated on the same subset. All RAG methods use BGE-large-en-v1.5 as the retrieval encoder and retrieve the top-5 sessions as context. All baselines use Qwen2.5-7B-Instruct as the inference backbone. More implementation details can be referred to Appendix C.2. Unless otherwise stated, all experiments in the main text are conducted on MUSE-128K, which is the setting all 14 models and 20 baselines share. MUSE-1M serves as a stress test of how performance degrades as the memory horizon grows, and is reported for eight long-context models in Appendix D.2.

[2]An industry memory framework without publication; https://github.com/langchain-ai/langmem.

Table 2: Main results of 14 frontier LLMs on **MUSE-Bench** under the full-context setting. Per column, marks the best and the second best; ties share the same rank.

Model	Overall		Personalized		Experience		User		Assistant		Interference		Counterfactual	
	ACC↑	HR↓	ACC↑	HR↓	ACC↑	HR↓	ACC↑	HR↓	ACC↑	HR↓	ACC↑	HR↓	ACC↑	HR↓
MiniMax-M2.5	0.6083	0.3185	0.9286	0.0132	0.7029	0.2495	0.3841	0.5198	0.7782	0.1585	0.6006	0.3239	0.1358	0.7755
Kimi-K2.5	0.6529	0.3419	0.9473	0.0527	0.8419	0.1581	0.3339	0.6520	0.8697	0.1303	0.8742	0.1164	0.1332	0.8642
Qwen3.5-plus	0.6036	0.3859	0.9341	0.0648	0.7733	0.2267	0.2811	0.7181	0.8504	0.1479	0.6384	0.2453	0.1462	0.8538
Qwen3.6-plus	0.6172	0.3617	0.9166	0.0812	0.7886	0.2114	0.3515	0.6441	0.8732	0.1268	0.4686	0.2987	0.2010	0.7990
Qwen3-MAX	0.6180	0.3414	0.9495	0.0494	0.8819	0.1181	0.2423	0.6581	0.8504	0.1408	0.7736	0.1258	0.1070	0.8799
GLM-4.7	0.6312	0.3500	0.9363	0.0571	0.7790	0.2171	0.3630	0.6264	0.8820	0.1109	0.4686	0.3899	0.2611	0.7311
GLM-5	0.6672	0.2914	0.9473	0.0450	0.8400	0.1371	0.3850	0.5674	0.8680	0.1127	0.5220	0.2799	0.4230	0.5457
Gemini-2.5-flash	0.6419	0.3448	0.9462	0.0461	0.8210	0.1733	0.3604	0.6247	0.8750	0.1144	0.6321	0.3208	0.1697	0.8225
Gemini-3-flash	0.6565	0.3211	0.9462	0.0494	0.7505	0.2476	0.3868	0.6132	0.8820	0.1162	0.4591	0.2987	0.4674	0.5248
Gemini-3.1-Flash-Lite	0.6240	0.3753	0.9232	0.0768	0.7829	0.2171	0.2837	0.7137	0.8539	0.1461	0.7421	0.2579	0.2637	0.7363
GPT-5.2	0.6130	0.3029	0.9429	0.0450	0.8057	0.1810	0.2634	0.5427	0.8926	0.0951	0.6195	0.1698	0.1802	0.7911
GPT-5.4	0.6404	0.3490	0.9396	0.0604	0.7924	0.2076	0.3295	0.6643	0.8697	0.1232	0.7013	0.2170	0.2507	0.7363
Claude Sonnet 4.6	0.6503	0.2862	0.9297	0.0472	0.8095	0.1676	0.3850	0.5031	0.8715	0.0986	0.7296	0.1164	0.1593	0.7937
Claude Opus 4.6	0.6698	0.2750	0.9341	0.0593	0.8286	0.1714	0.3815	0.5154	0.9067	0.0898	0.6541	0.1006	0.3394	0.6371
Average	*0.6353*	*0.3318*	*0.9373*	*0.0534*	*0.7999*	*0.1917*	*0.3379*	*0.6116*	*0.8659*	*0.1222*	*0.6346*	*0.2329*	*0.2313*	*0.7494*

3.2 Evaluating Large Language Models under the Full-Context Setting

To assess how well frontier LLMs leverage long-term memory, we evaluate 14 models on **MUSE-Bench** under the full-context setting. As reported in Table 2, we highlight the following findings:

- **LLMs' memory utilization remains at the understanding level, particularly for personalized information.** All models achieve high accuracy on personalized memory understanding, averaging 0.9373 with a hallucination rate of only 0.0534, indicating that surfacing explicit preferences from long context is largely within reach. However, when the target information is non-personalized and task-oriented, such as experience memory, accuracy drops to 0.7999, suggesting that requirements accumulated through task execution are harder to identify than stated preferences.
- **APPLY tasks expose a sharp capability gap.** When queries require applying rather than reporting memory, average accuracy falls to 0.3379. Two factors compound: the queries do not reference the relevant memory, making it harder to locate; and once located, it must still be combined with additional reasoning or world knowledge. Assistant memory application is far easier, as users reference assistant-side content explicitly and the tasks are typically revisions rather than reasoning.
- **Models fail confidently rather than abstaining.** Option E is available on every item, yet models select it on only 3.3% of questions, so hallucination rate nearly complements accuracy. The effect is starkest where accuracy is lowest: models abstain on only 5.1% and 1.9% of user memory application and counterfactual items. Failing to locate a memory and failing to recognize that it was not located are therefore distinct failures, and current models exhibit both together.
- **Interference resistance and counterfactual detection exhibit an adversarial trade-off.** Models that resist memory-based distractors well tend to perform poorly on counterfactual detection, and vice versa. No model achieves strong performance on both subtypes simultaneously, revealing a inherent conflict between trusting the query and trusting the memory.

3.3 Evaluating Memory Methods under the Retrieval-Based Setting

To evaluate how existing memory methods perform on **MUSE-Bench**, we benchmark 20 methods in Table 3 spanning three categories: context-utilization baselines, training-free memory methods, and trainable memory methods. All methods use Qwen2.5-7B-Instruct as the inference backbone.

Context-utilization baselines cluster tightly around 0.48–0.51 overall, with Full-Context the weakest: placing the entire history in the prompt is worse than retrieving a few relevant sessions. Training-free memory methods do not separate from them, and several fall below NaiveRAG; structured memory organization brings little benefit when the bottleneck is deciding whether a memory applies at all. Trainable methods achieve the two highest overall accuracies, and MemAgent is the only baseline to make headway on counterfactual evaluation, yet they trade this against personalized understanding, where every training-free method surpasses them. No category improves uniformly across ability levels. Hallucination rates tell a starker story: every method exceeds 0.44 overall, against 0.33 for frontier LLMs, so the 7B backbone not only answers wrong more often but abstains even less. Training-free memory methods improve only marginally over RAG-based baselines, suggesting that current memory techniques, particularly the training-free ones, contribute little to memory utilization.

Table 3: Main results of 20 memory methods on **MUSE-Bench**, the majority under a retrieval-based setting. All methods use the Qwen2.5-7B-Instruct backbone. marks the best and the second best.

Method	Overall		Personalized		Experience		User		Assistant		Interference		Counterfactual	
	ACC↑	HR↓	ACC↑	HR↓	ACC↑	HR↓	ACC↑	HR↓	ACC↑	HR↓	ACC↑	HR↓	ACC↑	HR↓
Full-Context	0.4323	0.5631	0.8738	0.1262	0.3876	0.6124	0.1860	0.8062	0.5278	0.4588	0.4252	0.5709	0.0428	0.9572
NaiveRAG	0.4798	0.5107	0.8988	0.0971	0.4402	0.5598	0.2237	0.7652	0.6036	0.3808	0.5866	0.3819	0.0296	0.9671
RAG-Fusion	0.4897	0.5028	0.9029	0.0943	0.4474	0.5526	0.2370	0.7508	0.6459	0.3452	0.5748	0.4016	0.0164	0.9836
InstructRAG	0.5080	0.4647	0.8849	0.0999	0.3732	0.6244	0.3555	0.6002	0.6080	0.3341	0.6181	0.3661	0.0132	0.9836
GenerativeAgents	0.4900	0.4815	0.9154	0.0721	0.4019	0.5981	0.2137	0.7121	0.6592	0.3229	0.6378	0.3504	0.0493	0.9507
MemGPT	0.4890	0.4982	0.8779	0.1193	0.4354	0.5646	0.2359	0.7309	0.6058	0.3898	0.7008	0.2795	0.0428	0.9572
MemoryBank	0.4808	0.4933	0.8890	0.1082	0.3565	0.6435	0.2558	0.6700	0.5924	0.3875	0.6378	0.3583	0.0559	0.9441
LangMem	0.4579	0.5297	0.8641	0.1290	0.3301	0.6699	0.2115	0.7697	0.5902	0.3875	0.6693	0.3071	0.0296	0.9704
Mem0	0.4628	0.5316	0.8877	0.1110	0.3278	0.6722	0.2248	0.7730	0.5612	0.4165	0.6457	0.3386	0.0493	0.9507
SimpleMem	0.4641	0.5241	0.8558	0.1387	0.3660	0.6340	0.2071	0.7641	0.6214	0.3697	0.6654	0.3268	0.0329	0.9671
Secom	0.4985	0.4884	0.9265	0.0693	0.4952	0.5048	0.2171	0.7575	0.6236	0.3675	0.6378	0.3228	0.0230	0.9770
A-Mem	0.4979	0.4936	0.9348	0.0610	0.4904	0.5096	0.2248	0.7608	0.6013	0.3898	0.6142	0.3622	0.0329	0.9671
HippoRAG2	0.4992	0.4874	0.9168	0.0804	0.4809	0.5191	0.2436	0.7298	0.6080	0.3742	0.6142	0.3583	0.0362	0.9638
MemGAS	0.5067	0.4861	0.9348	0.0624	0.4880	0.5120	0.2525	0.7398	0.6036	0.3786	0.6260	0.3543	0.0296	0.9704
MemoryOS	0.4831	0.5116	0.9390	0.0569	0.4569	0.5431	0.2193	0.7741	0.5702	0.4276	0.5354	0.4409	0.0493	0.9507
LightMem	0.4779	0.5061	0.8988	0.0957	0.4402	0.5598	0.2259	0.7442	0.5679	0.4053	0.6102	0.3701	0.0362	0.9605
Mem-α	0.4670	0.4949	0.8058	0.1942	0.2751	0.7249	0.2846	0.5925	0.5969	0.3920	0.7441	0.2559	0.0461	0.9539
MemGen	0.4739	0.4841	0.8433	0.1567	0.2608	0.7392	0.2846	0.5770	0.5657	0.4276	0.8110	0.1890	0.0362	0.9638
MemAgent	0.5362	0.4490	0.8460	0.1540	0.5096	0.4904	0.2702	0.6811	0.6860	0.3118	0.7559	0.2441	0.2237	0.7763
MemCoE	0.5412	0.4588	0.8433	0.1567	0.4402	0.5598	0.3688	0.6312	0.6860	0.3140	0.7480	0.2520	0.0888	0.9112
Average	*0.4868*	*0.4980*	*0.8870*	*0.1092*	*0.4102*	*0.5897*	*0.2471*	*0.7165*	*0.6062*	*0.3791*	*0.6429*	*0.3415*	*0.0482*	*0.9513*

3.4 Retrieval Quality of Memory Methods

Before any reasoning takes place, the relevant session has to be located at all. We isolate this stage by measuring retrieval quality for the thirteen methods that expose a retrieval step, reporting Recall and NDCG in Table 4. Retrieval is the binding constraint for most methods. At the top-5 cutoff actually used in our experiments, the best method recovers only 0.6980 of the relevant sessions, so roughly three items in ten enter generation without the memory they depend on, which places a ceiling on accuracy no downstream reasoning can lift. Structured memory organization helps, but unevenly: MemGAS and A-Mem lead at the top-3 and top-5 cutoffs, yet Recall@1 stays below 0.11 for every method. Aggressive abstraction is what hurts. Methods that compress sessions before indexing them, such as LightMem, fall below NaiveRAG at every cutoff, discarding precisely the surface detail that retrieval depends on.

Table 4: Retrieval quality of memory methods on **MUSE-Bench**. marks the best and the second best.

Method	Recall@1	NDCG@1	Recall@3	NDCG@3	Recall@5	NDCG@5
NaiveRAG	0.0541	0.1781	0.2896	0.3541	0.5877	0.4935
RAG-Fusion	0.0603	0.1982	0.3477	0.3934	0.6028	0.5141
InstructRAG	0.0859	0.2517	0.4382	0.4718	0.6828	0.5882
GenerativeAgents	0.1046	0.2568	0.2968	0.3942	0.3860	0.4382
MemoryBank	0.0600	0.2111	0.1922	0.3200	0.2227	0.3369
LangMem	0.0567	0.1558	0.3332	0.3683	0.5756	0.4880
Mem0	0.0695	0.1895	0.3299	0.3876	0.5585	0.4958
Secom	0.0830	0.2497	0.4375	0.4719	0.6874	0.5904
A-Mem	0.0977	0.2705	0.4562	0.4943	0.6937	0.6052
HippoRAG2	0.0774	0.2258	0.4129	0.4651	0.6914	0.5844
MemGAS	0.1023	0.2689	0.4651	0.5003	0.6980	0.6148
MemoryOS	0.0745	0.1922	0.2673	0.3402	0.4126	0.4154
LightMem	0.0394	0.1490	0.2083	0.2978	0.3201	0.3621

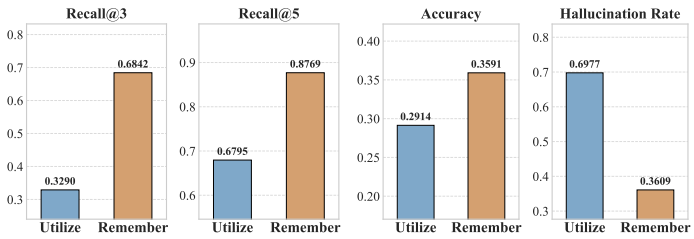
3.5 Time–Performance Trade-off of Memory Methods

Memory construction carries a cost that varies by orders of magnitude across methods, which accuracy alone does not reflect. Figure 4 plots overall accuracy against memory construction time on **MUSE-Bench**. No consistent relationship emerges. Trainable methods occupy the slow end: MemCoE and MemAgent are the two strongest but require hundreds of seconds, and Mem-α is slower still yet lands near the bottom. At the opposite extreme, InstructRAG and RAG-Fusion construct their memory in under a tenth of a second and still reach roughly 49–51%, within a few points of methods four orders of magnitude slower. Spending more on memory construction is therefore neither necessary nor sufficient; design matters more than budget.



Figure 4: The Time Costs for Memory Methods.

3.6 Why Is Utilization Harder?



(a) Retrieval Performance (b) Generation Performance

Figure 5: Performance on utilize versus remember queries.

To understand why user memory application and counterfactual evaluation exhibit such low accuracy, we devise the following experiment. For each answer-bearing memory, we prompt an LLM to generate a *remember* query that directly asks about the memory content from the user's perspective, and compare it against the original *utilize* query used in **MUSE-Bench**. Both are evaluated with BGE-large-en-v1.5 under session-level retrieval on **MUSE-Bench**. As shown in Figure 5, the gap opens at both stages. Retrieval comes first: remember queries score substantially higher, confirming that need-driven queries are harder to retrieve for because they do not reference the target memory. Generation compounds it. Accuracy on utilize queries remains lower even where retrieval succeeds, but the sharper contrast is in hallucination rate, which nearly doubles. Asked a factual question it cannot answer, a model will often say so; asked instead to carry out a task, it carries the task out, filling the missing memory with a plausible substitute rather than reporting that it is absent. The failure is therefore twofold: models struggle to locate the relevant memory, and having failed to locate it, they do not recognize the failure.

4 Related Work

4.1 Memory Benchmark

Existing memory benchmarks fall along two complementary axes. LongMemEval (Wu et al., 2025) and LoCoMo (Maharana et al., 2024) focus on memory recall in long multi-session dialogues, evaluating whether models can retrieve and reason over information from extended chat histories. Their queries explicitly point to the target memory, framing memory as the answer to be retrieved. PersonaMem (Jiang et al., 2025a;b) shifts toward implicit personalization, evaluating whether models can infer evolving user profiles from interaction history and select personalized responses. While PersonaMem moves beyond pure recall by requiring inference of latent preferences, it remains confined to preference-based response selection and assumes memory should always be invoked. Across all three benchmarks, the question of *whether and when* memory should be used remains unaddressed, which is central to deployment scenarios where memory may be irrelevant, misleading, or contradicted by the current query. **MUSE-Bench** fills this gap by systematically evaluating three higher-order memory abilities, Understand, Apply, and Evaluate, under a unified Bloom-inspired taxonomy, with particular emphasis on the Evaluate level's bidirectional invocation decisions.

4.2 Memory Technique

Methods for equipping LLM agents with long-term memory broadly fall into two categories. RAG-based methods such as MemGPT (Packer et al., 2023), MemoryBank (Zhong et al., 2024), A-MEM (Xu et al., 2025b), and Mem0 (Chhikara et al., 2025) treat memory as a retrievable external store with varying structures and retrieval strategies, offering modularity but limited adaptability when query–memory relevance is non-trivial. RL-based methods such as Memory-R1 (Yan et al., 2025) train adaptive memory operations through reinforcement learning, achieving greater flexibility at the cost of training overhead and limited generalization. Both lines of work focus on optimizing memory storage and retrieval, while leaving the higher-order utilization abilities evaluated in MUSE-Bench largely uncharacterized. A more comprehensive discussion is provided in Appendix F.

5 Conclusion

We introduced **MUSE-Bench**, a benchmark for evaluating higher-order memory utilization beyond basic recall, grounded in a Bloom-inspired taxonomy of Understand, Apply, and Evaluate. Our evaluation of 14 frontier LLMs and 20 memory methods reveals that current models perform reasonably at the Understand level but exhibit substantial gaps at the Apply and Evaluate levels, where they neither locate the memory a task depends on nor recognize when they have failed to locate it. Existing memory techniques offer limited improvement, and these findings suggest that the next frontier of agent memory lies not in memorizing more, but in using memory more wisely.

AI USE STATEMENT

The **MUSE-Bench** construction pipeline is LLM-driven, as documented in Section 2 and Appendix B. GPT-5.2 generates all personas, needs, memories, queries, and dialogue histories; Gemini-3-flash serves as an independent auditor at several stages; and GPT-5.2 is also used to annotate the WildChat pilot study in Appendix A. No single model both generates and approves an item, and all flagged items are additionally reviewed by human annotators. We report inter-annotator agreement and pass rates throughout so that the reliability of AI-generated content can be assessed directly. Separately, we used generative AI to polish the writing of this paper. The research questions, benchmark design, taxonomy, experimental protocol, and all analyses are our own, and every AI-assisted output has been reviewed by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Both the evaluation code and the dataset are anonymously released. The code is available at https://anonymous.4open.science/r/MUSE-Bench-FC48/README.md and the dataset at https://huggingface.co/datasets/anonymous111111111/MUSE-Bench. The five-stage construction pipeline is described in Section 2, with per-stage filters, distractor definitions, and human validation protocols in Appendix B, and the related prompts in Appendix G. Model versions, API snapshots, and the shared evaluation prompt template are given in Section 3 and Appendix C.2, which also lists retrieval encoders, RL training hyperparameters, and hardware. Persona generation uses fixed random seeds, and evaluation is multiple-choice with deterministic scoring, so results can be reproduced exactly from the released data.

REFERENCES

Qingyao Ai, Yichen Tang, Changyue Wang, Jianming Long, Weihang Su, and Yiqun Liu. Memorybench: A benchmark for memory and continual learning in llm systems. *arXiv preprint arXiv:2510.17281*, 2025.

Lorin W Anderson and David R Krathwohl. *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives: complete edition*. Addison Wesley Longman, Inc., 2001.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2023.

Sanghwan Bae, Donghyun Kwak, Soyoung Kang, Min Young Lee, Sungdong Kim, Yuin Jeong, Hyeri Kim, Sang-Woo Lee, Woomyoung Park, and Nako Sung. Keep me updated! memory management in long-term conversations. *arXiv preprint arXiv:2210.08750*, 2022.

Yuanchen Bei, Tianxin Wei, Xuying Ning, Yanjun Zhao, Zhining Liu, Xiao Lin, Yada Zhu, Hendrik Hamann, Jingrui He, and Hanghang Tong. Mem-gallery: Benchmarking multimodal long-term conversational memory for mllm agents. *arXiv preprint arXiv:2601.03515*, 2026.

Haonan Bian, Zhiyuan Yao, Sen Hu, Zishan Xu, Shaolei Zhang, Yifu Guo, Ziliang Yang, Xueran Han, Huacan Wang, and Ronghao Chen. Realmem: Benchmarking llms in real-world memory-driven interaction. *arXiv preprint arXiv:2601.06966*, 2026.

Okan Bursa. A dynamic retrieval-augmented generation system with selective memory and remembrance. *arXiv preprint arXiv:2601.02428*, 2026.

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

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

Yuntian Deng, Wenting Zhao, Jack Hessel, Xiang Ren, Claire Cardie, and Yejin Choi. Wildvis: Open source visualizer for million-scale chat logs in the wild. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 497–506, 2024.

Yiming Du, Hongru Wang, Zhengyi Zhao, Bin Liang, Baojun Wang, Wanjun Zhong, Zezhong Wang, and Kam-Fai Wong. Perltqa: A personal long-term memory dataset for memory classification, retrieval, and fusion in question answering. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pp. 152–164, 2024.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, et al. Lightmem: Lightweight and efficient memory-augmented generation. *arXiv preprint arXiv:2510.18866*, 2025.

Xueyang Feng, Weinan Gan, Xu Chen, Quanyu Dai, and Yong Liu. How does personalized memory shape llm behavior? benchmarking rational preference utilization in personalized assistants. *arXiv preprint arXiv:2601.16621*, 2026.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. 2024.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: neurobiologically inspired long-term memory for large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pp. 59532–59569, 2024.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*, 2025.

Zexue He, Yu Wang, Churan Zhi, Yuanzhe Hu, Tzu-Ping Chen, Lang Yin, Ze Chen, Tong Arthur Wu, Siru Ouyang, Zihan Wang, et al. Memoryarena: Benchmarking agent memory in interdependent multi-session agentic tasks. *arXiv preprint arXiv:2602.16313*, 2026.

Rebecca MM Hicke and Kiran Tomlinson. Adopt $\neq$ adapt: Longitudinal analyses of llm conversations in the wild. *arXiv preprint arXiv:2605.29018*, 2026.

Matthew Ho, Chen Si, Zhaoxiang Feng, Fangxu Yu, Yichi Yang, Zhijian Liu, Zhiting Hu, and Lianhui Qin. Arcmemo: Abstract reasoning composition with lifelong llm memory. *arXiv preprint arXiv:2509.04439*, 2025.

Chuanrui Hu, Tong Li, Xingze Gao, Hongda Chen, Dannong Xu, Yi Bai, Tianwei Lin, Xinda Zhao, Xiaohong Li, Jiaqi An, et al. Evermembench: Benchmarking long-term interactive memory in large language modelsevermembench: Benchmarking long-term interactive memory in large language models. *arXiv preprint arXiv:2602.01313*, 2026a.

Yulin Hu, Zimo Long, Jiahe Guo, Xingyu Sui, Xing Fu, Weixiang Zhao, Yanyan Zhao, and Bing Qin. Op-bench: Benchmarking over-personalization for memory-augmented personalized conversational agents. *arXiv preprint arXiv:2601.13722*, 2026b.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, et al. Memory in the age of ai agents. *arXiv preprint arXiv:2512.13564*, 2025.

Bowen Jiang, Zhuoqun Hao, Young-Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo J Taylor, and Dan Roth. Know me, respond to me: Benchmarking llms for dynamic user profiling and personalized responses at scale. *arXiv preprint arXiv:2504.14225*, 2025a.

Bowen Jiang, Yuan Yuan, Maohao Shen, Zhuoqun Hao, Zhangchen Xu, Zichen Chen, Ziyi Liu, Anvesh Rao Vijjini, Jiashu He, Hanchao Yu, et al. Personamem-v2: Towards personalized intelligence via learning implicit user personas and agentic memory. *arXiv preprint arXiv:2512.06688*, 2025b.

Cheng Jiayang, Dongyu Ru, Lin Qiu, Yiyang Li, Xuezhi Cao, Yangqiu Song, and Xunliang Cai. Amemgym: Interactive memory benchmarking for assistants in long-horizon conversations. In *The Fourteenth International Conference on Learning Representations*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory os of ai agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25972–25981, 2025.

Seo Hyun Kim, Tzu-iunn Ong, Taeyoon Kwon, Namyoung Kim, Keummin Ka, SeongHyeon Bae, Yohan Jo, Seung-won Hwang, Dongha Lee, Jinyoung Yeo, et al. Theanine: Revisiting memory management in long-term conversations with timeline-augmented response generation. *arXiv e-prints*, pp. arXiv–2406, 2024.

Johannes Kirmayr, Lukas Stappen, Phillip Schneider, Florian Matthes, and Elisabeth Andre. Carmem: Enhancing long-term memory in llm voice assistants through category-bounding. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pp. 343–357, 2025.

Hung Le, Dai Do, Dung Nguyen, and Svetha Venkatesh. Reasoning under 1 billion: Memory-augmented reinforcement learning for large language models. *arXiv preprint arXiv:2504.02273*, 2025.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. A human-inspired reading agent with gist memory of very long contexts. In *International Conference on Machine Learning*, pp. 26396–26415. PMLR, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Hao Li, Chenghao Yang, An Zhang, Yang Deng, Xiang Wang, and Tat-Seng Chua. Hello again! llm-powered personalized agent for long-term dialogue. *arXiv preprint arXiv:2406.05925*, 2024.

Jinze Li, Yang Zhang, Xin Yang, Jiayi Qu, Jinfeng Xu, Shuo Yang, Junhua Ding, and Edith Cheuk-Han Ngai. Ocr-memory: Optical context retrieval for long-horizon agent memory. *arXiv preprint arXiv:2604.26622*, 2026a.

Xinze Li, Ziyue Zhu, Siyuan Liu, Yubo Ma, Yuhang Zang, Yixin Cao, and Aixin Sun. Emembench: Interactive benchmarking of episodic memory for vlm agents. *arXiv preprint arXiv:2601.16690*, 2026b.

Yuyang Li, Yime He, Zeyu Zhang, and Dong Gong. Evimem: Evidence-gap-driven iterative retrieval for long-term conversational memory. *arXiv preprint arXiv:2604.27695*, 2026c.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. Simplemem: Efficient lifelong memory for llm agents. *arXiv preprint arXiv:2601.02553*, 2026a.

Zeyuan Liu, Jeonghye Kim, Xufang Luo, Dongsheng Li, and Yuqing Yang. Exploratory memory-augmented llm agent via hybrid on-and off-policy optimization. *arXiv preprint arXiv:2602.23008*, 2026b.

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239*, 2023.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 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)*, pp. 13851–13870, 2024.

Niloofar Mireshghallah, Neal Mangaokar, Narine Kokhlikyan, Arman Zharmagambetov, Manzil Zaheer, Saeed Mahloujifar, and Kamalika Chaudhuri. Cimemories: A compositional benchmark for contextual integrity in llms. In *The Fourteenth International Conference on Learning Representations*.

Kai Tzu-iunn Ong, Namyoung Kim, Minju Gwak, Hyungjoo Chae, Taeyoon Kwon, Yohan Jo, Seungwon Hwang, Dongha Lee, and Jinyoung Yeo. Towards lifelong dialogue agents via timeline-based memory management. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8631–8661, 2025.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Jianfeng Gao. Secom: On memory construction and retrieval for personalized conversational agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xKDZAW0He3.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023a.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023b.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023c.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM on Web Conference 2025*, pp. 2366–2377, 2025.

Zackary Rackauckas. Rag-fusion: a new take on retrieval-augmented generation. *arXiv preprint arXiv:2402.03367*, 2024.

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

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=GN921JHCRw.

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

Yiting Shen, Kun Li, Wei Zhou, and Songlin Hu. Mem2actbench: A benchmark for evaluating long-term memory utilization in task-oriented autonomous agents. *arXiv preprint arXiv:2601.19935*, 2026b.

Juntao Tan, Liangwei Yang, Zuxin Liu, Zhiwei Liu, Rithesh RN, Tulika Manoj Awalgaonkar, Jianguo Zhang, Weiran Yao, Ming Zhu, Shirley Kokane, et al. Personabench: Evaluating ai models on understanding personal information through accessing (synthetic) private user data. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 878–893, 2025.

Mohammad Tavakoli, Alireza Salemi, Carrie Ye, Mohamed Abdalla, Hamed Zamani, and J Ross Mitchell. Beyond a million tokens: Benchmarking and enhancing long-term memory in llms. *arXiv preprint arXiv:2510.27246*, 2025.

Juyuan Wang, Rongchen Zhao, Wei Wei, Yufeng Wang, Mo Yu, Jie Zhou, Jin Xu, and Liyan Xu. Comorag: A cognitive-inspired memory-organized rag for stateful long narrative reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33557–33565, 2026a.

Qingyue Wang, Yanhe Fu, Yanan Cao, Shuai Wang, Zhiliang Tian, and Liang Ding. Recursively summarizing enables long-term dialogue memory in large language models. *Neurocomputing*, pp. 130193, 2025a.

Yu Wang and Xi Chen. Mirix: Multi-agent memory system for llm-based agents. *arXiv preprint arXiv:2507.07957*, 2025.

Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng Li, Xian Li, Bing Yin, et al. Memoryllm: Towards self-updatable large language models. In *International Conference on Machine Learning*, pp. 50453–50466. PMLR, 2024a.

Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. Mem-{\alpha}: Learning memory construction via reinforcement learning. *arXiv preprint arXiv:2509.25911*, 2025b.

Zheng Wang, Zhongyang Li, Zeren Jiang, Dandan Tu, and Wei Shi. Crafting personalized agents through retrieval-augmented generation on editable memory graphs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4891–4906, 2024b.

Zhenting Wang, Huancheng Chen, Jiayun Wang, and Wei Wei. Memex (rl): Scaling long-horizon llm agents via indexed experience memory. *arXiv preprint arXiv:2603.04257*, 2026b.

Zhepei Wei, Wei-Lin Chen, and Yu Meng. Instructrag: Instructing retrieval-augmented generation via self-synthesized rationales. In *International Conference on Learning Representations*, volume 2025, pp. 82731–82754, 2025.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pZiyCaVuti.

Tingyu Wu, Zhisheng Chen, Ziyan Weng, Shuhe Wang, Chenglong Li, Shuo Zhang, Sen Hu, Silin Wu, Qizhen Lan, Huacan Wang, et al. Knowme-bench: Benchmarking person understanding for lifelong digital companions. *arXiv preprint arXiv:2601.04745*, 2026.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101, 2025.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, Tong Xu, et al. From single to multi-granularity: Toward long-term memory association and selection of conversational agents. *arXiv preprint arXiv:2505.19549*, 2025a.

Derong Xu, Shuochen Liu, Pengfei Luo, Pengyue Jia, Yingyi Zhang, Yi Wen, Yimin Deng, Wenlin Zhang, Enhong Chen, Xiangyu Zhao, and Tong Xu. Learning how and what to memorize: Cognition-inspired two-stage optimization for evolving memory, 2026. URL https://arxiv.org/abs/2605.00702.

Jing Xu, Arthur Szlam, and Jason Weston. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2022.

Wujiang Xu, Kai Mei, Hang Gao, Juntao Tan, Zujie Liang, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*, 2025b.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Hinrich Schütze, Volker Tresp, and Yunpu Ma. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *arXiv preprint arXiv:2508.19828*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, et al. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *arXiv preprint arXiv:2507.02259*, 2025.

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. Agentic memory: Learning unified long-term and short-term memory management for large language model agents. *arXiv preprint arXiv:2601.01885*, 2026.

Guibin Zhang, Muxin Fu, and Shuicheng Yan. Memgen: Weaving generative latent memory for self-evolving agents. *arXiv preprint arXiv:2509.24704*, 2025a.

Zeyu Zhang, Quanyu Dai, Luyu Chen, Zeren Jiang, Rui Li, Jieming Zhu, Xu Chen, Yi Xie, Zhenhua Dong, and Ji-Rong Wen. Memsim: A bayesian simulator for evaluating memory of llm-based personal assistants. *arXiv preprint arXiv:2409.20163*, 2024.

Zeyu Zhang, Quanyu Dai, Rui Li, Xiaohe Bo, Xu Chen, and Zhenhua Dong. Learn to memorize: Optimizing llm-based agents with adaptive memory framework. *arXiv preprint arXiv:2508.16629*, 2025b.

Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. Do llms recognize your preferences? evaluating personalized preference following in llms. In *13th International Conference on Learning Representations, ICLR 2025*, pp. 72531–72574. International Conference on Learning Representations, ICLR, 2025a.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatgpt interaction logs in the wild. In *International Conference on Learning Representations*, volume 2024, pp. 34590–34605, 2024.

Yujie Zhao, Boqin Yuan, Junbo Huang, Haocheng Yuan, Zhongming Yu, Haozhou Xu, Lanxiang Hu, Abhilash Shankarampeta, Zimeng Huang, Wentao Ni, et al. Ama-bench: Evaluating long-horizon memory for agentic applications. *arXiv preprint arXiv:2602.22769*, 2026.

Zheng Zhao, Clara Vania, Subhradeep Kayal, Naila Khan, Shay B Cohen, and Emine Yilmaz. Personalens: A benchmark for personalization evaluation in conversational ai assistants. *arXiv preprint arXiv:2506.09902*, 2025b.

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

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. *arXiv preprint arXiv:2506.15841*, 2025.

Technical Appendix

TABLE OF CONTENTS

A WildChat Experiments **17**

A.1 Pilot Study Pipeline 17

A.2 Final Statistics 18

A.3 WildChat Case Study 18

B Benchmark Construction Details **21**

B.1 Persona Construction 21

B.2 Real User Need Construction 21

B.3 Memory Construction 24

B.4 Need-to-Query Realization 25

B.5 Multiple-Choice Option Construction 25

B.6 Dialogue Synthesis 25

B.7 Human Validation 26

C Baselines and Implementation **26**

C.1 Baselines Details. 26

C.2 Implementation Details 27

D Addition Experiments **28**

D.1 Domain-Level Performance 28

D.2 Detailed LLM Performance on MUSE-1M 28

D.3 Ablation Study of Memory Context 30

E Benchmark Analysis **31**

E.1 Dataset Statistics 31

E.2 Case Study 31

E.3 Cost Analysis 32

F Related Work **33**

F.1 Memory Benchmark 33

F.2 RAG-based Memory Techniques 36

F.3 RL-based Memory Techniques 36

G Prompts **36**

A WILDCHAT EXPERIMENTS

To understand how prior sessions are relied upon in real human-AI interactions, we conduct a pilot study on **WildChat** (Zhao et al., 2024; Deng et al., 2024), a public corpus of 3.2M ChatGPT conversations collected in the wild, covering everyday tasks such as coding assistance, writing, planning, and casual conversation. Each conversation carries a hashed IP address, a request header, and a UTC timestamp, which together let us reconstruct and temporally order the sessions of a single user. For queries that draw on the user's interaction history, we ask how that memory is used: *whether it is a simple act of recalling stored facts, or a more complex form of utilization: reasoning over scattered memories, inferring implicit preferences, or judging whether a memory applies at all, among others.*

A.1 PILOT STUDY PIPELINE

Answering this question requires deciding, for every query in the sample, whether it draws on the user's history at all, a judgment that cannot be reduced to keyword matching, since surface markers are unreliable in both directions. We therefore build a user-level sample, judge each query against that user's own history, and classify how that history is used, in the following three stages.

Stage 1: Building User-Level Sample. Following Hicke & Tomlinson (2026), we group conversations into pseudo-users and discard those whose activity pattern is inconsistent with a single returning person, such as shared or rotating addresses and automated traffic, since these would contaminate a user's history with content belonging to others. This leaves 2.5M conversations from 1.8M pseudo-users. Since memory use can only arise where a history exists, we then restrict the pool to users with more than five conversations and randomly sample 1,000 of them, yielding 15,462 conversations and 35,979 user queries.

Stage 2: Memory Detection. We use GPT-5.2 (`gpt-5.2-2025-12-11`) to decide whether answering a candidate query requires the user's history, and to identify the sessions that supply it. The criterion is comparative rather than lexical: a session counts as used if an assistant holding it would answer better than one that had never seen it, which admits dependence on past content as well as on the user's standing preferences, language, and expected output format. To keep each verdict verifiable rather than asserted, the model must name every session it relies on and, where the dependence is on content, quote a verbatim span from that session. Naming the sessions also narrows what the next stage sees. The memory detection prompt is provided in Figure 16.

Stage 3: Memory Utilization Classification. On the verified memory-using queries, we again employ GPT-5.2 to assign one of two labels. A query is labelled **A. Recall** when the retained sessions already contain the substance of a correct response, so that answering amounts to returning that content, either verbatim or re-expressed through translation, condensation, or reformatting. A query

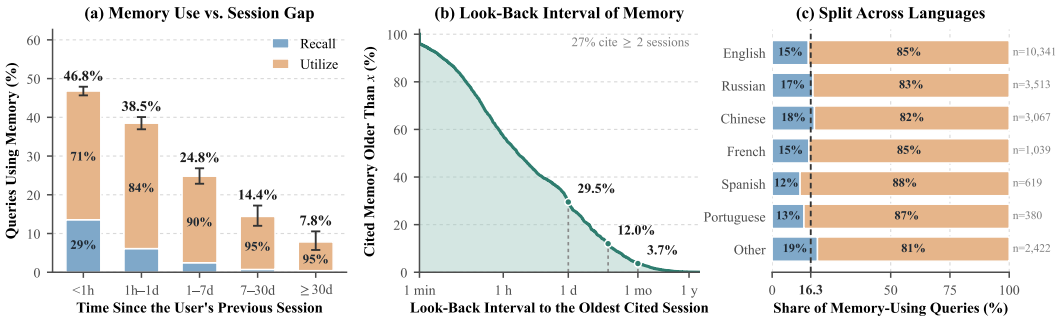


Figure 6: Memory use in WildChat. **(a)** Share of session-opening queries that use memory, binned by the interval since the user's previous session and split into RECALL and UTILIZE. Whiskers are 95% Wilson intervals. **(b)** Among the memory-using queries that cite at least one earlier session, the share whose oldest cited session lies further back than x. **(c)** RECALL–UTILIZE split by language; the dashed line marks the overall recall share.

is labelled **B. Utilize** when a correct response cannot be derived from the retained content alone, so that memory serves as a premise for further work rather than as the answer itself. This category subsumes reasoning over information distributed across earlier sessions, conforming to preferences, conventions, and standing instructions the user established previously, and judging which memories still bear on the present request while setting aside those that are outdated or merely superficially similar. The full annotation prompt for Stage 3 is provided in Figure 17.

A.2 Final Statistics

Applied to the 35,979 candidate queries of the sampled users, the pipeline yields 21,382 verified memory-using queries (59.4%). Of these, 16.3% directly ask about past content, while the remaining 83.7% use it as a substrate for a new request.

Figure 6 (a) shows how this behavior varies with elapsed time. Memory use becomes less frequent as the interval since the user's previous session grows, and what remains shifts increasingly toward Utilize. This follows from what the two categories require: content from the past hour is recent enough to be worth restating, whereas content from weeks earlier survives in the conversation only as a background condition for whatever the user is doing now. Memory nonetheless reaches well beyond the immediately preceding exchange, as shown in Figure 6 (b): among queries that cite an earlier session, close to a third look back more than a day, and over a quarter draw on several sessions.

The 16.3% split is not an artifact of the annotation criterion or of any single language. It remains at 16.4% under the permissive criterion, and the recall share stays between 12% and 18% across the corpus's major languages (Figure 6 (c)). In ordinary requests, the dependence on prior sessions is predominantly one of premise for new work rather than of retrieval for its own sake, and increasingly so the further back the memory lies, a finding that directly motivates the benchmark design in Section 2.2.

To assess the reliability of the GPT-5.2 annotations, we randomly sampled 500 queries from the verified set and had two annotators, both graduate students in natural language processing, independently label each as Recall or Utilize without access to the model's predictions. The two annotators agreed on 94.8% of queries (Cohen's $\kappa = 0.787$), and each agreed with GPT-5.2 on 92.2% and 91.8% ($\kappa = 0.674$ and 0.662). Where the annotators reached consensus, that consensus matched GPT-5.2 on 94.3% of queries, and all three recall shares fell within one percentage point of one another, consistent with a boundary that does not depend on any single rater. Disagreements were concentrated on borderline queries that combine both modes, such as a request to restate a prior response and then extend it.

Moreover, to analyze the composition of *utilize* instances, we sampled 1,000 such queries and classified each into three cognitive-demand levels: Understand, Apply, and Evaluate, by prompting GPT-5.2 with the rubric in Figure 19, followed by manual verification. The resulting distribution is Understand 59.9%, Apply 30.7%, and Evaluate 9.4%, indicating that while the majority of memory use involves straightforward comprehension of stated preferences or context, a substantial ∼40% requires non-trivial inference or selective judgment about which memory to trust.

A.3 WildChat Case Study

Figures 7 and 8 present two instances of each label. In R1 the information a correct answer requires is stored in a reply given twenty-four minutes earlier, and in R2 the query refers back to figures the assistant has just produced. In both, everything a correct reply must state is already present in the retained sessions, so answering reduces to locating that content and returning it. *In a conversational setting, the session holding the answer usually shares the topic and much of the wording of the current query, so even basic representation-based retrieval locates it with a high success rate.* This is the setting existing memory benchmarks are built around.

The Utilize cases place a higher demand on how memory is used. In U1 the answer is not contained anywhere in the history; what the history supplies is the user's standing preference for short answers, which the model must first infer from behaviour that is never stated as a rule and then apply to a request about a novel the history never mentions. In U2 the history supplies the task definition, the current state of the code, and an alternative the user has already rejected, and the model must reason across these sessions and decide which of them still bears on the request before any of the architecture

CASE R1 in WildChat

DIALOGUE HISTORY

[S11] *24 minutes before the query*

U I've set up a static library in premake5 and built it. Can other projects use it without setting header paths...?

A Yes. Another project only has to link the library and include its headers...

U Is there a way to do it like Visual Studio's "reference" — just tick the project you depend on, and the header and lib paths get set for you?

A You can use `dependson` to declare the dependency between projects, then `links` and `includedirs` to set the library and header paths...

U What I mean is, **once `dependson` is set, do I still need `includedirs`?**

A **In general you do not need to set `includedirs` by hand**: the dependency's header paths are propagated to the main project automatically...

CURRENT QUERY

U With premake5, once `dependson` is set, is `includedirs` not needed?

ANNOTATION

RECALL The last exchange of [S11] put and answered this question, so the reply need only return it.

CASE R2 in WildChat

DIALOGUE HISTORY

[S2] *2.3 hours before the query*

U what is the full nutritional breakdown of horseradish **powder** (it must be powder) per 100g

A I don't have specific information for horseradish powder. Here is horseradish **root**... Calories: 48; Carbohydrates: 11 grams...

U also all the phytonutrients

A I don't have a list specific to the powder... here are phytonutrients commonly found in horseradish **root**...

[S3] *minutes before the query*

U what is the nutritional breakdown of **dried and powdered** horseradish root including phytonutrients

A **Here is an approximate breakdown per 100 grams of dried and powdered horseradish root**: Calories: 48; Carbohydrates: 11 grams...

CURRENT QUERY

U were *those figures* for the raw root or the dried and powdered root

ANNOTATION

RECALL [S3] already labels the figures as dried and powdered root, which is exactly the query asks.

Figure 7: Two representative RECALL instances in WildChat.

choice or the implementation can be produced. Recall asks where the answer is stored; UTILIZE asks what follows from what is stored.

This is also why the two differ in difficulty. Under RECALL the query names the content it needs, which makes both retrieving that content and verifying the answer against it straightforward. Under UTILIZE the query names a new need instead, so the relevant memory shares little surface form with it and must be found without a lexical anchor, and once found it still has to be reasoned over, interpreted, or judged for applicability. Retrieval and utilization thus compound rather than substitute, which is what Section 2.2 is designed to measure and what Figure 5 confirms quantitatively.

CASE U1 in WildChat

DIALOGUE HISTORY

[S5] *17 days before*

A 1. She wakes up to the soothing sound of birds chirping outside, her room bathed in warm sunlight, ready to face the day ahead...

U **Remove all the commas from each line**

[S8] *7 days before*

A In the dimly lit expanse of a colossal underground cavern, an eerie aura hung in the air, unveiling a mesmerizing scene of an ancient and enigmatic massive pyramid...

U **remove all the commas from that sentence**

[S10] *2 days before*

U write a sentence describing a looming ominous unseen creature in the darkness against an even creepier nightmare setting in the form of **a stable diffusion prompt**

A As the moonless night grew colder and thick tendrils of fog rolled in, an unseen, malevolent creature roamed stealthily in the darkness...

U **remove the commas from that sentence**

A As the moonless night grew colder and thick tendrils of fog rolled in an unseen malevolent creature roamed stealthily in the darkness...

CURRENT QUERY

U describe an eerie haunted graveyard with phantom optical illusions dancing throughout in a single sentence in the form of a **stable diffusion prompt**

ANNOTATION

UTILIZE The user works with Stable Diffusion prompts, where the comma is a delimiter, so a description cannot use commas as punctuation. This requirement is never stated as such. It surfaces only in correction issued after each draft: [S5], [S8], and [S10], every sentence the assistant produces is followed by a request to strip the commas. The current query names only the scene and says nothing about the constraint.

CASE U2 in WildChat

DIALOGUE HISTORY

[S3] *2 days before the query*

U Part I — Building the Environment. I implemented independent ROS nodes ... a node that enables the car to scan its environment using a laser range finder ...

U Implement the complete evader function ... For the reward function ... in the worst case scenario if the car hits an obstacle or gets too close to one ...

[S8] *19.9 hours before*

A I would suggest employing the Proximal Policy Optimization (PPO) algorithm rather than DQN for this scenario.

U Thx for you suggestion! But let's stick to our original implementation.

CURRENT QUERY

U Now Implement the end to end DQNAgentPytorch that fit our task. Choose the best model for this kind of scenario and implement an end to end DQN Agent.

ANNOTATION

UTILIZE [S3] supplies the task and the code as it stands and [S8] rules out PPO, but the architecture choice and the implementation itself are new work.

Figure 8: Two representative UTILIZE instances in WildChat.

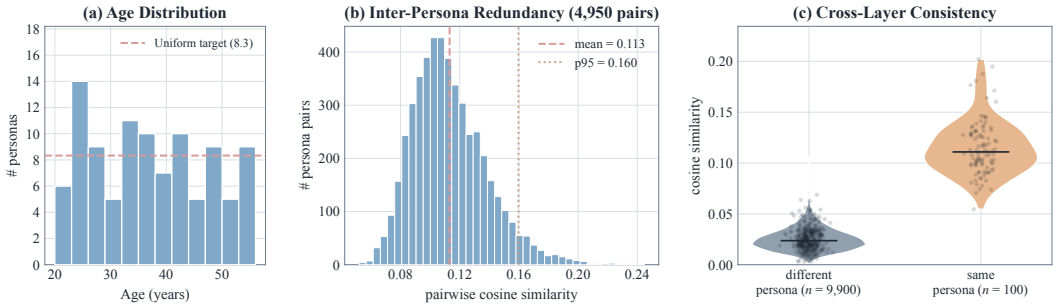


Figure 9: Diversity and internal coherence of the 100 constructed personas. **(a)** Age matches the intended uniform target; **(b)** pairwise similarity reveals no near-duplicate personas; **(c)** a persona's preference and experience layers stay far closer to each other than to those of any other persona.

B Benchmark Construction Details

B.1 Persona Construction

We start by building a *personal skeleton* for each user, in three layers. The basic information layer (name, age, gender, occupation) is generated deterministically using the Python `Faker` library with fixed random seeds, so that the persona pool is reproducible across runs. As Fig. 9(a) shows, the age histogram tracks the uniform reference line across the whole 20–55 range, so the pool is not concentrated in any single life stage; similarly, the per-occupation counts in Fig. 10 stay close to a flat allocation over ten professional sectors, and even the most frequent title is held by only seven users, so no single profession dominates. The LLM then enriches this skeleton with two further layers: a user preference layer that articulates the user's stable likes, habits, and routines across daily-life scenarios, and a user experience layer that captures the user's profession-related procedural knowledge and long-running responsibilities. Because both layers are conditioned on the same skeleton, the pool is diverse across users while remaining coherent within each user. Fig. 9(b) measures the lexical similarity of every pair of personas: the distribution concentrates in a narrow low band and even the closest pair stays below 0.25, so two users seldom share the same tastes or habits and no persona degenerates into a near-duplicate of another. Fig. 9(c) contrasts the two enriched layers taken from the same user against those taken from different users; the same-user distribution sits more than four times higher and barely overlaps the different-user one, which indicates that a user's preferences and professional experience read as descriptions of one and the same person rather than two independently sampled profiles. The resulting personal skeleton is the single anchor that conditions every subsequent generation step, and is shared across all task scenarios so that one persona corresponds to *one coherent user lifetime* rather than disjoint synthetic fragments.

B.2 Real User Need Construction

Given a persona, we first synthesize the need and then reverse-engineer the minimal set of supporting memories, ensuring that each memory is indispensable for resolving the query.

B.2.1 Understand Layer.

This layer evaluates whether an assistant has internalized the memory it has built up about a user over long-term interaction. Crucially, such understanding is not confined to personalization. Existing benchmarks (e.g., PersonalMem, PrefEval) are built around modeling user preferences—including how those preferences persist and shift over long horizons—with the aim of producing a personalized response. Yet a long-term collaborator must grasp not only what the user prefers, but also how their everyday tasks are actually carried out. When the same kind of task recurs and the same requirement is imposed on it each time, that regularity likewise reveals what the user needs from a given task. Such requirements often have nothing to do with preference. More often, the user is themselves constrained—by an employer, a client, an institutional rule, or the setting in which the work is done—and must therefore impose particular demands on the format or the content of what is

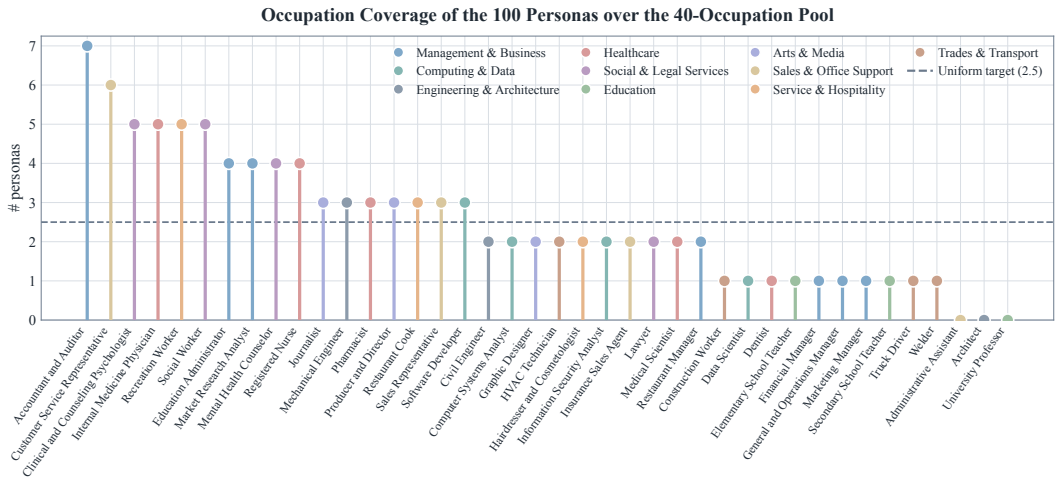


Figure 10: Occupation coverage of the 100 personas over the 40-occupation pool, coloured by sector. Sampled counts stay close to the uniform target of 2.5, with 37 of the 40 titles covered and a maximum load of seven personas.

produced. We therefore decompose this layer into **personalized needs**, which concerns what the user prefers, and **experience needs**, which concerns the requirements their tasks must satisfy.

When the same kind of task recurs and the same requirement is imposed on it each time, that regularity likewise reveals what the user needs from a given task. Such requirements often have nothing to do with preference: they are constraints the user is themselves under

Personalized needs. Each need must be something the user genuinely cannot settle on their own and would realistically bring to an assistant—and one that the user's personalized context allows to be served better. Existing datasets tend to probe such context mechanically, asking the assistant outright what it knows about the user rather than situating that knowledge within a need, so an assistant that passes those probes may still fail to bring the same knowledge to bear once a real request arrives. A user would never ask what kind of seat they prefer, since that much is already settled for them; but they may well ask the assistant to handle a flight booking, and it is the assistant's grasp of that preference that turns an adequate booking into a well-judged one. To cover the diverse ways such familiarity manifests, we construct needs around six complementary aspects of that context—stable preferences, habitual routines, recurring cycles, physical and health constraints, emotional patterns, and relational facts about people in the user's life—across the twelve life domains introduced above. For each user we sample a subset of domains and construct multiple needs per domain, so that no single aspect or topic dominates the resulting distribution.

Experience needs. Needs of this kind capture the requirements a user brings to the format or the content of everyday tasks. Such requirements are typically not of the user's own choosing; they arise instead from external or situational constraints—an employer's template, a client's specification, an institutional or regulatory rule—which the user is expected to observe. They are rarely stated as explicit preferences. More often they accumulate through ordinary working exchanges, in the corrections a user issues and the clarifications they supply after seeing a draft. The question is then whether an assistant, on meeting a comparable request later on, can draw on those earlier constraints to produce a more appropriate result. Asking it outright what format the user's firm expects would settle nothing; what matters in practice is whether the assistant applies the relevant constraint of its own accord while carrying the task out—whether a follow-up email drafted after a case meeting arrives with the case reference and the confidentiality disclaimer already in place. We construct such needs around two complementary aspects of these requirements—how an output should be structured, and what it must contain to be judged adequate—again across the twelve life domains, so that the constraints a user works under remain consistent with their occupation and setting.

Verification. Because our construction is need-driven, the need is the basepoint from which every downstream artifact, including memories, queries, and distractors, is derived, and a flawed need

silently propagates into all of them. We therefore submit every instance to an independent audit by Gemini-3-flash, applying one check common to both categories and one specific to each. The common check concerns plausibility: whether the need is one a person would actually raise in the course of ordinary life, rather than an artifact of construction. For personalized needs, we additionally check for conflicts among the preferences and habits a need draws on, since these are elicited along separate dimensions and need not cohere: a user described as drinking black coffee every morning cannot also be one who avoids caffeine altogether, and a need resting on both is unusable. For experience needs, we check instead that each recorded requirement is stated unambiguously: a rule must carry a single requirement, with no compound conditions and no unresolved references, so that compliance can be adjudicated rule by rule. Of 1,200 personalized instances, 237 were removed and 963 retained; of 600 experience instances, 49 were removed and 551 retained, leaving 1,514 of the 1,800 originally constructed. A random sample of the retained set is then inspected manually: 300 instances (190 personalized and 110 experience, sampled proportionally across schemas and domains) are judged by a single annotator, yielding an overall pass rate of 98.0% (97.9% personalized, 98.2% experience).

B.2.2 APPLY LAYER.

Apply shifts the question from what an assistant has come to understand of a user to whether that understanding is actually summoned when a task calls for it. This layer evaluates whether, when a need arising in everyday use depends on remembered content as a necessary step toward its answer, the assistant recognizes as much of its own accord and then recalls and applies that content correctly. Existing benchmarks largely put their questions to memory directly—how long the user's commute is, when they last travelled—which can show that the relevant content has been recalled, but not whether the assistant would draw on it unprompted. Real needs seldom take that form. A user who knows the distance from home to the office would never ask it; but when their check-in time is moved up, they may well ask when they ought to set out—a question answerable only by bringing that remembered fact together with the new circumstance. The need does not itself ask about anything held in memory, yet it cannot be resolved accurately without it. Recall and application are in this sense distinct capacities, and the distance between them is considerable. The memory drawn upon may originate with the user, or with the assistant itself, whose earlier lists, plans, and schedules likewise become the basis of what is asked next. We therefore divide this layer into **user-related needs** and **assistant-related needs**, according to whose record the task must be carried out against.

User-related needs. Needs of this kind are likewise ones a user might genuinely raise in the course of everyday life, except that some piece of information about the user is a necessary input to resolving them. The remembered material is put to work in two ways: through quantitative or deductive reasoning over what the request newly supplies, or by pairing it with common knowledge anyone could contribute—a unit conversion, a time-zone offset, a standard formula. We construct three situations in which this dependence arises. In the first, a single remembered fact suffices, and the assistant need only bring it together with the circumstance at hand. In the second, no single fact will do: several must be combined, and the answer becomes underivable if any one of them is absent. The third we treat as a category of its own because it turns on time rather than content—the answer depends on when things were recorded, how events stand relative to one another, or how a situation has changed since, and must be worked out against a deadline or interval the request introduces.

Assistant-related needs. What the assistant must draw on is not confined to what it has learned about the user; its own earlier replies serve equally well as memory. Over the course of collaboration the assistant produces content of its own—a plan laid out in steps, for instance—which the user may later want adjusted or improved upon. Such requests fall into three kinds, according to what is to be done with that content: completing it, where the user asks for the part still outstanding; comparing it, where the user brings something of their own and asks how the two stand against each other; and revising it, where an element is to be removed or exchanged and the updated version returned. In none of these cases does the user restate what the assistant said before; whether that content can be correctly recalled and put to use is precisely what the need demands.

Verification. The constructed needs are again submitted to a cross-model audit, along two lines. The first is naturalness: whether the need reads as a situation a person would actually find themselves in. The second is necessity: whether the remembered content is genuinely required as a step in deriving the answer. If the request can be resolved from its own wording alone, the memory is doing no work

and the instance is discarded. Of 1,200 user-related instances, 27 were removed and 1,173 retained; of 600 assistant-related instances, 16 were removed and 584 retained, leaving 1,757 of the 1,800 originally constructed. We further draw a stratified random sample of 300 instances (200 user-related and 100 assistant-related, proportional to each subset's size) for manual inspection, finding a pass rate of 97.0% (194/200) for user-related needs and 99.0% (99/100) for assistant-related needs.

B.2.3 EVALUATE LAYER.

Evaluation turns from whether remembered content is brought to bear to whether it ought to be. The other two scenarios reward drawing on memory, and an assistant that recalls accurately and puts what it holds to use will fare well on them—yet this disposition is not always the right one. What is remembered may be genuine, current, and squarely on topic, and still not serve the case at hand. This layer therefore evaluates whether an assistant weighs what it holds against the situation before it, rather than deferring by default either to memory or to the request. Two situations bring this out. In the first, the record is accurate and even matches the topic the request raises, yet it does not fit the context in which the request is made: a user's standing preference is for rock music and the question is indeed about what to put on, but the occasion is a formal reception, where that preference is precisely what should not be followed—what is measured is whether the assistant applies a stated preference blindly. In the second, the user asserts something about their own circumstances that the record contradicts: a request asks for help arranging a friend's birthday celebration in June, while memory has that friend's birthday in July—what is measured is whether the assistant simply accepts the premise it is given, or recognises the discrepancy and says so. We accordingly divide this layer into **interference needs**, where memory interferes with the situation the request presents, and **counterfactual needs**, where memory conflicts with the situation as the request describes it.

Interference needs. A need of this kind places a recorded preference or habit in a setting whose conventions do not align with it. The preference and the setting must bear on the same dimension, so that applying what is remembered produces a concrete answer that is wrong for the case rather than merely beside the point. Following the preference stays available throughout—the setting renders it inappropriate, never impossible—and the convention it must yield to is one nearly any reader would recognise without being told. The distractors are drawn accordingly: each is what an assistant would arrive at by honouring the preference faithfully.

Counterfactual needs. In a need of this kind, what the request states about the user's situation contradicts what the record says about it. The contradicted detail is always an objective fact—the make or capacity of something the user owns—rather than something a person could settle by simple decision, and it must be the premise that decides the answer: what follows from the request as stated and what follows from the record must each be concrete and must differ. An assistant that takes the user at their word therefore lands on an answer that is identifiably wrong, and what the need measures is whether the mismatch is noticed and raised at all.

Verification. We again submit the constructed needs to cross-model validation. The audit first checks whether the scenario is natural enough that a person would plausibly encounter it. For interference needs, it additionally verifies that the situational norm overriding the preference is common-sense rather than niche, and that the preference and the situational expectation act on the same decision dimension so that following the preference produces a concretely wrong choice rather than merely an irrelevant one. For counterfactual needs, it verifies that the contradicted detail is a hard fact rather than a mutable preference, and that the detail is load-bearing: the answer derived from the request and the answer derived from the record must be concretely different. Of 400 interference instances, 63 were removed, leaving 337; of 400 counterfactual instances, 7 were removed, leaving 393, totalling 730 of the 800 originally constructed. A stratified random sample of 200 instances (100 interference, 100 counterfactual) was inspected by a single annotator, yielding pass rates of 96.0% and 98.0% respectively.

B.3 MEMORY CONSTRUCTION

For each accepted need, we first reverse-derive the minimum set of *answer-bearing memories* required to produce the correct response. These memories are intentionally sparse: only what is strictly necessary to disambiguate the question. If we stopped here, the resulting history would

be unrealistically thin and trivially keyword-retrievable, allowing models to bypass long-context reasoning.

We therefore additionally synthesize a layer of *scaffolding memories*: persona-consistent background memories (life updates, secondary preferences, ongoing tasks, relational facts) that are *not* required to answer any specific query, but together compose the texture of a real user's long-running life. The answer-bearing memories are embedded inside this scaffold, so that retrieving them at test time is a search problem rather than a matching problem. Across needs and across users, all generated memories are then run through a cross-consistency pass that detects and resolves mutual contradictions before they enter the dialogue stage.

B.4 Need-to-Query Realization

A need expresses an abstract intent; a *query* is the actual first-person utterance the user sends to the assistant at a future timestamp $t_q > t_N$. We render every need into such a query and apply two filters before accepting it. The *leakage filter* rejects queries whose surface text re-states or strongly hints at the answer-bearing memory, since otherwise the model could solve the item without using its memory at all. The *disambiguation filter* rejects queries that are simultaneously consistent with two or more user memories, ensuring each item has a unique persona-grounded answer.

B.5 Multiple-Choice Option Construction

We frame every item as a 5-way multiple-choice question rather than open-ended generation. The choice is deliberate. First, at 1M-token context, evaluating open-ended answers requires LLM-as-judge calls whose cost scales with both the number of items and the length of the rendered prompt; multiple-choice scoring is deterministic and reproducible. Second, lexical metrics (BLEU/ROUGE/F1) are known to be unreliable for personalized responses, where the difference between a correct and an incorrect answer often lies in *which user fact was used*, not in surface wording. Third, multiple-choice options give us a precise instrument to test against *specific failure modes* of long-context personalized assistants, which we exploit in distractor design below.

Each item carries one gold answer and three hard-negative distractors, drawn from four canonical failure modes:

- **Other-persona preference distractor.** The answer is plausible in general but is grounded in a preference taken from a *different* persona; testing whether the model uniquely binds memory to the current user.
- **Outdated-memory distractor.** The answer applies an earlier, since-updated version of the same user's preference or fact; testing whether the model performs memory *update* rather than first-mention recall.
- **Premise-following distractor.** The answer blindly accepts a counterfactual premise embedded in the query itself (used in conflict-false items); testing whether the model can override a deceptive query with established memory.
- **Generic-default distractor.** A reasonable, domain-default answer that consults no user-specific memory at all; testing whether the model is genuinely using personal history rather than its parametric prior.

Because every distractor is itself grounded in something *about the user* (or about a sibling user), all four options are surface-plausible, and trivial pattern matches (e.g., longest option, repeated entities) are not exploitable.

B.6 Dialogue Synthesis

The memories produced in Section B.3 are not exposed directly to the model; they must be revealed through multi-turn *dialogue*. We render each memory into a topic-coherent conversation between the user and the assistant: explicit memories are stated by either party in plain text, while implicit memories are surfaced only through the user's choices, behaviors, and preferences as they unfold across turns. To prevent the resulting history from being a thinly disguised concatenation of relevant facts, we additionally inject a substantial volume of *filler sessions*—small talk, off-topic requests,

and unrelated assistance interactions—so that retrieving the answer-bearing content requires actual long-context memory rather than scanning a short relevant window. All sessions are timestamped along a multi-month timeline using a recency-skewed sampler, and interleaved across topics. Finally, the *same* memory and query set is re-woven at two different filler densities to produce the 128K-token (**MUSE-128K**) and 1M-token (**MUSE-1M**) variants of the benchmark, enabling controlled comparison across context budgets while holding the underlying questions fixed.

B.7 Human Validation

To verify the quality of the final benchmark, we conducted a systematic human evaluation with three independent annotators, all holding PhD degrees in computer science with research experience in dialogue systems and natural language understanding.

We randomly sampled 500 items from the released 3,840, stratified across all six query types with heavier sampling of the Evaluate layer (roughly 130, 160, and 210 items for the Understand, Apply, and Evaluate layers respectively). Each item was independently assessed by all three annotators along five criteria: *i) Answer Correctness*: whether the gold answer is correct given the full conversation history; *ii) Distractor Plausibility*: whether all distractors are surface-plausible yet definitively incorrect, with no ambiguous or partially-correct options; *iii) Query Naturalness*: whether the query reads as a natural user utterance and is free of answer leakage that could allow solving without memory; *iv) Dialogue Coherence*: whether the synthesized conversation history is topically coherent, stylistically consistent, and free of abrupt transitions or contradictions between turns; *v) Memory Grounding*: whether the answer-bearing memories are faithfully and unambiguously embedded within the dialogue context, such that the gold answer is derivable solely from the conversation history.

An item passes only when at least two annotators independently approve. The inter-annotator agreement, measured by Fleiss' κ, reached 0.87 across the five dimensions, indicating near-perfect consensus. The overall pass rate is 96.4% (482/500). Per-criterion unanimous-pass rates are: Answer Correctness 98.6%, Distractor Plausibility 97.4%, Query Naturalness 99.0%, Dialogue Coherence 98.6%, and Memory Grounding 97.8%. Breaking down by layer, the Understand layer achieves 98.5% (128/130), the Apply layer 96.9% (157/162), and the Evaluate layer 94.7% (197/208). Among the 18 flagged items, the most common issue is distractor plausibility (8 items), followed by answer correctness under edge-case temporal reasoning (5 items) and minor dialogue coherence gaps (5 items).

C Baselines and Implementation

C.1 Baselines Details.

We evaluate our benchmark with the following baselines:

- **NaiveRAG** (Lewis et al., 2020) retrieves relevant passages from an external corpus via dense vector search and prepends them to the LLM's context for answer generation.
- **RAG-Fusion** (Rackauckas, 2024) expands the original query into multiple related sub-queries, retrieves documents for each of them, and merges the resulting ranked lists via reciprocal rank fusion (RRF).
- **InstructRAG** (Wei et al., 2025) instructs the LM to self-synthesize rationales explaining how the answer derives from retrieved documents, enabling explicit denoising without additional supervision.
- **Generative Agents** (Park et al., 2023c) store an agent's experiences as a natural-language memory stream, synthesize them into higher-level reflections, and retrieve them dynamically to plan believable behavior.
- **MemGPT** (Packer et al., 2023) borrows virtual context management from operating systems, paging information between an in-context working memory and external storage tiers to extend the effective context window.
- **MemoryBank** (Zhong et al., 2024) equips LLMs with long-term memory that summarizes past interactions and user personality, updating and forgetting entries via an Ebbinghaus-inspired mechanism.

- **LangMem** provides memory tools that let agents record and search memories in the hot path, plus a background manager that extracts, consolidates, and updates them over time.
- **Mem0** (Chhikara et al., 2025) dynamically extracts, consolidates, and retrieves salient information from conversations via a scalable memory-centric architecture, with an optional graph-based variant for capturing relational structures.
- **SimpleMem** (Liu et al., 2026a) compresses interactions into multi-view indexed memory units, synthesizes related context online, and infers search intent to plan retrieval scope efficiently.
- **SeCom** (Pan et al., 2025) constructs a memory bank at the topical-segment level using a conversation segmentation model and applies compression-based denoising to improve retrieval accuracy.
- **A-Mem** (Xu et al., 2025b) organizes memories in an agentic way by following Zettelkasten principles, generating structured notes with contextual descriptions, keywords, and inter-memory links upon each new memory insertion.
- **HippoRAG2** (Gutiérrez et al., 2024; 2025) augments dense retrieval with a knowledge graph and Personalized PageRank to support factual, sense-making, and associative memory tasks simultaneously.
- **MemoryOS** (Kang et al., 2025) manages agent memory through an OS-inspired hierarchical architecture of short-term, mid-term, and long-term storage, with dynamic promotion policies (FIFO and segmented-page strategies) governing transitions between levels.
- **LightMem** (Fang et al., 2025) organizes memory into three Atkinson-Shiffrin-inspired stages—sensory, short-term, and long-term—using lightweight compression to filter redundancy while balancing performance and computational efficiency.
- **MemGAS** (Xu et al., 2025a) constructs multi-granularity memory units (sessions, summaries, keywords) and uses a Gaussian Mixture Model for memory association and an entropy-based router to adaptively select the optimal granularity per query.
- **MemAgent** (Yu et al., 2025) trains an LLM to read text in segments and update a fixed-length memory via an overwrite strategy, optimized end-to-end with a multi-conversation extension of the DAPO reinforcement learning algorithm.
- **Mem-α** (Wang et al., 2025b) trains agents via RL to construct a tri-component memory (core, episodic, semantic) using specialized tools for update, delete, merge, and restructure operations, with rewards derived from downstream QA performance.
- **MemGen** (Zhang et al., 2025a) equips agents with a *memory trigger* that decides when to invoke memory and a *memory weaver* that generates latent token sequences as machine-native memory, enabling a tightly interwoven cycle of memory and reasoning without altering model parameters.
- **MemCoE** (Xu et al., 2026) decouples memory organization from memory content via a two-stage framework: first inducing a transferable natural-language guideline through contrastive textual gradients, then optimizing a guideline-aligned memory evolution policy with multi-turn GRPO.

C.2 Implementation Details

We evaluate four RL-based memory methods on our benchmark: MemAgent, Mem-α, MemGen, and MemCoE. All methods use Qwen2.5-7B-Instruct as the base language model. For data partitioning, we adopt a user-level split: the first 20% of users (person_0$\sim$person_19) serve as the training set, while the remaining 80% (person_20$\sim$person_99) are held out for evaluation. All methods are trained for a unified budget of 100 steps, with validation performed every 10 steps. The reward function across all methods compares the predicted option label against the ground truth, yielding a binary 0/1 score. In addition to this QA reward, Mem-α incorporates two auxiliary reward terms: a compression reward (β=0.05) that penalizes redundant memory storage, and a content-classification reward (γ=0.1) that encourages information to be stored in the appropriate memory type (semantic vs. episodic).

Table 5 summarizes the key training hyperparameters and context strategies of each method. All four methods use Group Relative Policy Optimization (GRPO), a Critic-free algorithm that estimates advantages through intra-group comparison of multiple rollouts per prompt. For context processing, MemGen relies on an external retrieval service built with FAISS indexes and BGE-large-en-v1.5 embeddings, retrieving the top-3 relevant sessions per search query across multiple interaction turns. MemAgent and MemCoE adopt a chunk-based reading strategy, splitting all concatenated

Table 5: Training hyperparameters and context strategies of the four RL-based memory methods.

	MemAgent	Mem-α	MemGen	MemCoE
RL Algorithm	GRPO	GRPO	GRPO	GRPO
Learning Rate	1×10^{-6}	1×10^{-6}	1×10^{-6}	1×10^{-6}
KL Coefficient	0.001	0.001	0.0	0.001
Clip Ratio	0.2	0.2	—	0.2
Batch Size	16	16	16	16
Group Size N	4	4	4	4
Param. Update	Full	Full	LoRA	Full
Context Strategy	Chunk	Func. Call	Retrieval	Chunk

sessions into fixed-size chunks (5000 and 512 tokens, respectively) and iteratively updating a textual memory summary. Mem-α takes a distinct approach by operating a structured three-layer memory system (Core, Semantic, Episodic) through function calls, with a separate model serving as the question-answering backend.

Regarding parameter efficiency, MemAgent and MemCoE perform full-parameter optimization during RL training, while MemGen only trains LoRA adapters (r=16) along with its Memory Weaver and Trigger modules, keeping the base Reasoner frozen. All experiments are conducted on 8×H20 GPUs, employing FSDP for distributed training in veRL-based methods and DeepSpeed ZeRO-2 for MemGen.

D Addition Experiments

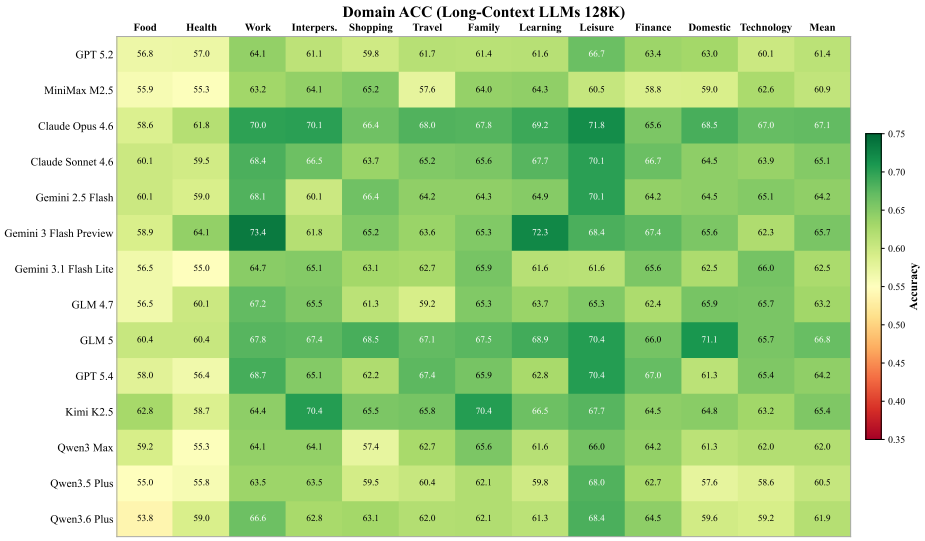
D.1 Domain-Level Performance

Long-Context LLMs. As shown in Figure 11a, we evaluate 14 long-context LLMs across 12 life domains on **MUSE-Bench**. Claude Opus 4.6 achieves the most balanced and consistently high performance, leading the mean column at 67.1% with its peak accuracy of 71.8% in Leisure. Gemini 3 Flash Preview attains the highest single-domain score overall (73.4% in Work) and also excels in Learning (72.3%), yet shows relatively weaker results in Interpersonal and Food. GLM 5 stands out in Domestic (71.1%) and Shopping (68.5%), demonstrating strength in practical daily-life domains. Kimi K2.5 is notably competitive in socially oriented domains, achieving 70.4% in both Interpersonal and Family. Across all models, Work, Leisure, and Learning consistently emerge as the easier domains, while Food and Health remain the most challenging, suggesting that LLMs handle structured professional and educational knowledge better than sensory or lifestyle-specific details.

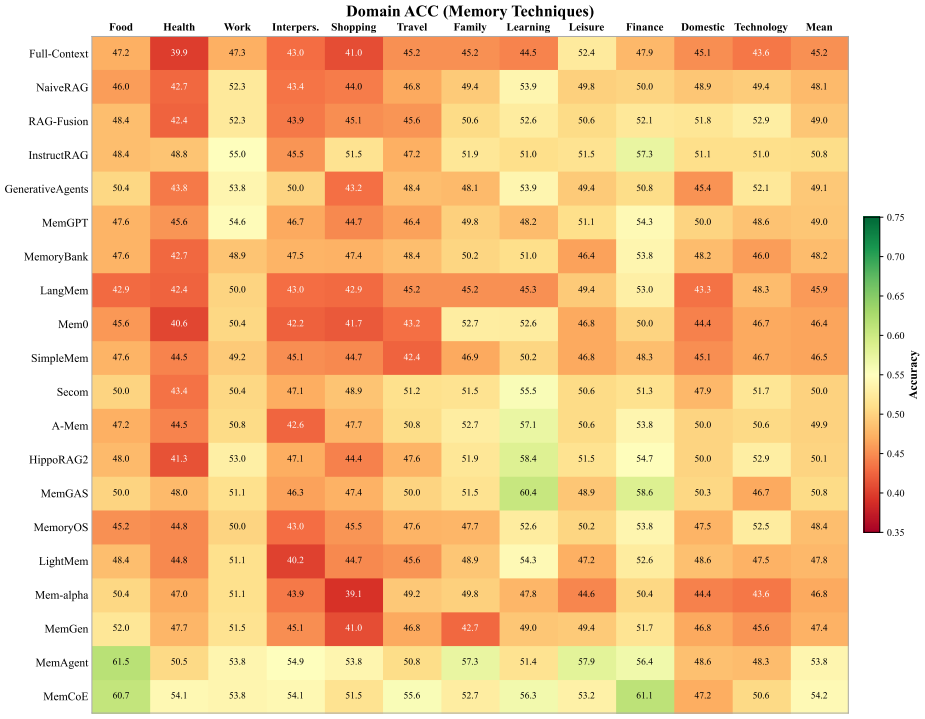
Memory Methods. As shown in Figure 11b, we compare 20 memory methods on **MUSE-Bench**. MemAgent achieves the broadest domain coverage, substantially outperforming other methods in Food (61.5%), Interpersonal (54.9%), and Leisure (57.9%), indicating its advantage in capturing diverse everyday knowledge. MemCoE leads the mean column at 54.2%, topping Health (54.1%), Travel (55.6%), and Finance (61.1%), the highest single-domain score among all baselines. MemGAS and HippoRAG2 both excel in Learning, at 60.4% and 58.4%, benefiting from their structured indexing over knowledge-intensive domains. In contrast, LangMem and Mem0 occupy the lower bound across nearly all domains, and Full-Context is the weakest overall at 45.2%, dropping to 39.9% in Health. Notably, most methods perform better in Work and Learning than in Shopping and Interpersonal, indicating that current memory techniques are more effective at retaining structured factual knowledge than tracking transactional or social details.

D.2 Detailed LLM Performance on MUSE-1M

Table 6 reports the full per-subtype results of 8 frontier LLMs on **MUSE-1M**. Scaling the context from 128K to 1M tokens leads to a consistent decline across all subtypes, with overall accuracy dropping from 0.6339 to 0.5894 on average. We summarize the detailed findings below.



(a) Long-Context LLMs (128K).



(b) Memory Techniques.

Figure 11: Domain-level accuracy on MUSE-BENCH.

- **Understanding remains the strongest capability but degrades unevenly.** Personalized memory understanding drops from 0.9373 to 0.8852 on average, while experience memory understanding drops from 0.7999 to 0.7383. GPT-5.4 retains the strongest understanding performance with 0.9517 on personalized and 0.7867 on experience subtypes. Qwen3.6-Plus suffers the largest decline, falling to 0.7849 on personalized understanding, the only model below 0.85 on this subtype.
- **User memory application remains the hardest subtype.** The average accuracy on user memory application falls from 0.3379 to 0.2939. Notably, Claude Sonnet 4.6 drops to 0.1965, the lowest

Table 6: Performance of 8 Frontier LLMs on **MUSE-1M** across query types.

Model	Overall ACC	Understand		Apply		Evaluate	
		Personalized	Experience	User	Assistant	Interference	Counterfactual
Qwen3.5-Plus	0.5682	0.8540	0.7143	0.2837	0.7359	0.7358	0.1436
Qwen3.6-Plus	0.5534	0.7849	0.6724	**0.3533**	0.6919	0.6101	0.1802
Gemini-2.5-Flash	0.5953	0.8826	0.7371	0.3242	0.8415	0.6352	0.1227
Gemini-3-flash	**0.6188**	0.9089	0.7333	0.3383	0.8539	0.4560	**0.3890**
Gemini-3.1-Flash-Lite	0.5924	0.8793	0.7333	0.2581	0.7870	0.7044	0.3264
GPT-5.4	0.6062	**0.9517**	**0.7867**	0.2564	**0.8627**	0.7138	0.1044
Claude Sonnet 4.6	0.5721	0.9100	0.7829	0.1965	0.7711	**0.8616**	0.0574
Claude Opus 4.6	0.6089	0.9100	0.7467	0.3410	0.8345	0.6352	0.1410
Average	0.5894	0.8852	0.7383	0.2939	0.7973	0.6690	0.1831

among all evaluated models, while Qwen3.6-Plus achieves the highest score of 0.3533. Assistant memory application also degrades from 0.8660 to 0.7973, with Gemini-2.5-Flash leading at 0.8415.

- **The adversarial trade-off between evaluation subtypes becomes more pronounced.** The tension between interference resistance and counterfactual detection sharpens at the 1M scale. Claude Sonnet 4.6 achieves the highest interference evaluation score of 0.8616 but scores only 0.0574 on counterfactual evaluation, the lowest among all models. Conversely, Gemini-3-Flash leads counterfactual evaluation at 0.3890 but shows the weakest interference resistance at 0.4560.

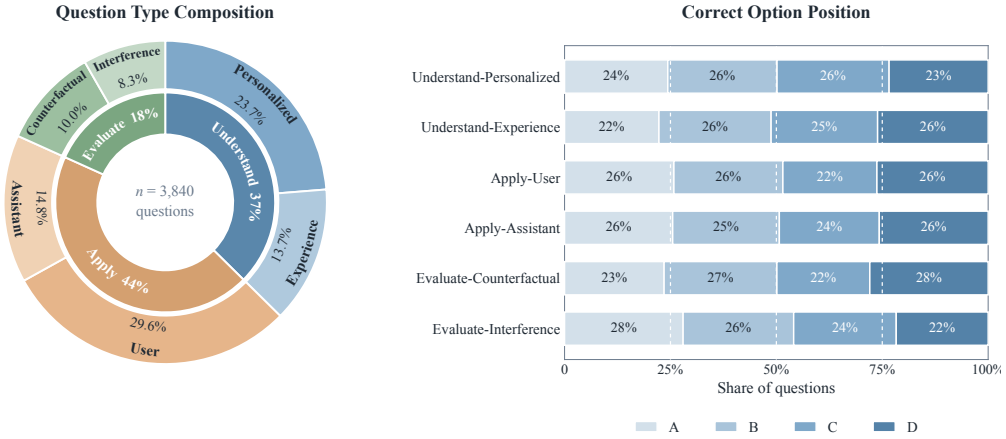
- **Gemini-3-Flash achieves the best overall performance.** With an overall accuracy of 0.6188, Gemini-3-Flash leads at the 1M scale, benefiting from a more balanced profile across subtypes. It is the only model that achieves above 0.33 on both user memory application and counterfactual evaluation.

D.3 Ablation Study of Memory Context

The validity of any multiple-choice benchmark rests on the premise that its options cannot be resolved from surface cues alone: if a model could recover the correct answer without ever consulting the dialogue history, high accuracy would measure commonsense priors rather than memory. To verify this, we evaluate two memory-free baselines that receive only the query and its options, with the entire interaction history withheld—Qwen2.5-7B-Instruct as a mid-sized instruction-tuned model and GPT-5.2 as a frontier model—and report the results in Table 7. Both baselines stay close to the 0.2000 random-guess floor (0.2776 and 0.2299 overall), and on A-User and A-Assistant queries, whose answers exist only in the history, GPT-5.2 collapses to 0.0079 and 0.0581 while abstaining on 97.0% and 93.8% of items, confirming that these questions are genuinely unanswerable rather than merely hard. The two baselines nevertheless fail in markedly different ways: GPT-5.2 hallucinates less than half as often as Qwen2.5-7B-Instruct (0.1992 versus 0.4776 HR) because it retreats to the abstention option in 57.1% of cases instead of committing to an unsupported choice, and it therefore attains a *lower* overall accuracy than the far smaller model—a counterintuitive ordering that reflects superior calibration rather than weaker competence, and that confirms the explicit "I don't know" option functions as an effective abstention channel. The memory-free setting further allows us to isolate the causal effect of adversarial memories. E-Interference queries are answered easily once their distracting sessions are removed (0.8774 for GPT-5.2 and 0.7956 for Qwen2.5-7B-Instruct), yet supplying that history drives accuracy down to 0.6195 and 0.4252, relative degradations of 29.4% and 46.6%; since the questions themselves are unchanged, this drop cannot stem from intrinsic difficulty and must instead originate from the interfering content, demonstrating that our distractors genuinely mislead memory systems rather than being harmlessly ignored. The one category with clearly above-chance memory-free accuracy is U-Personalized (0.4007 and 0.4204), which is expected since preference questions admit partial commonsense priors; the gap to the above-0.80 accuracy that memory-augmented systems achieve on this category nonetheless confirms that it still predominantly measures memory rather than prior knowledge.

Table 7: Accuracy (ACC) and hallucination rate (HR) of LLMs answering without any memory context.

Method	LLM	Overall		Personalized		Experience		User		Assistant		Interference		Counterfactual	
		ACC	HR	ACC	HR	ACC	HR	ACC	HR	ACC	HR	ACC	HR	ACC	HR
Random Guess	–	0.2000	0.6000	0.2000	0.6000	0.2000	0.6000	0.2000	0.6000	0.2000	0.6000	0.2000	0.6000	0.2000	0.6000
w/o Memory	Qwen2.5-7B-Instruct	0.2776	0.4776	0.4007	0.5664	0.2571	0.7429	0.0775	0.1859	0.3556	0.5194	0.7956	0.2013	0.0601	0.9347
	GPT-5.2	0.2299	0.1992	0.4204	0.0494	0.3333	0.6438	0.0079	0.0220	0.0581	0.0035	0.8774	0.0377	0.0104	0.8956



(a) Question type composition.

(b) Correct option position.

Figure 12: Question types and answer layout.

E Benchmark Analysis

E.1 Dataset Statistics

We analyse the composition and the answer-key balance of the benchmark to verify that it covers diverse reasoning demands while offering no shortcut that could be exploited without memory.

Figure 12a shows that the 3,840 questions span three reasoning levels—Understand (37%), Apply (44%), and Evaluate (18%)—each split into two categories, with no single category exceeding 30%. The benchmark therefore probes a broad spectrum of memory skills rather than one dominant pattern.

Figure 12b shows that the correct option is placed almost uniformly: every category keeps each position within 22–28%, the overall distribution is indistinguishable from uniform ($\chi^2(3) = 2.99$, $p = 0.39$), and the position is independent of question type ($\chi^2(15) = 13.24$, $p = 0.58$). Position priors thus give no advantage, and accuracy must come from the retrieved evidence.

E.2 Case Study

The aggregate results in Section 3 report where models fail, but not what the failures look like. To characterise them, we inspect items on which frontier LLMs agree on a wrong option, since unanimity indicates a systematic disposition rather than noise. We present three such cases, drawn from the subtypes with the lowest accuracy: user memory application (Figure 13), assistant memory application (Figure 14), and counterfactual evaluation (Figure 15).

The failures are not failures of reasoning or recall in the ordinary sense. In Figure 13 the required arithmetic is a single division, and no model gets it wrong; what all 14 models get wrong is which of two recorded measurements is still in force, choosing the older value because its session matches the query almost word for word while the current one matches only in substance. Surface similarity, in other words, overrides recency. Figure 14 shows what happens when retrieval fails outright: half the models return the three lounges the user just named, unchanged. Had they located the earlier recommendation and merely mis-computed the intersection, they would have landed on a partial

Table 8: Summary of the evaluated large language models, including the provider, exact API snapshot, release date, context window, and official per-million-token pricing (USD).

Provider	Model	API Snapshot / Version	Release	Context	Input	Output
OpenAI	GPT-5.4	`gpt-5.4-2026-03-05`	2026-03-05	1,050K	$2.50	$15.00
	GPT-5.2	`gpt-5.2-2025-12-11`	2025-12-11	400K	$1.75	$14.00
Anthropic	Claude Opus 4.6	`claude-opus-4-6` (2026-02-05)	2026-02-05	1,000K	$5.00	$25.00
	Claude Sonnet 4.6	`claude-sonnet-4-6-20260217`	2026-02-17	1,000K	$3.00	$15.00
Google	Gemini 3 Flash	`gemini-3-flash-preview`	2025-12-17	1,000K	$0.50	$3.00
	Gemini 3.1 Flash-Lite	`gemini-3.1-flash-lite-preview`	2026-03-03	1,000K	$0.25	$1.50
	Gemini 2.5 Flash	`gemini-2.5-flash`	2025-06-17	1,000K	$0.30	$2.50
Alibaba	Qwen3-Max	`qwen3-max-2026-01-23`	2026-01-23	262K	$1.20	$6.00
	Qwen3.6-Plus	`qwen3.6-plus` (2026-04-02)	2026-04-02	1,000K	$0.50	$3.00
	Qwen3.5-Plus	`qwen3.5-plus-2026-02-15`	2026-02-15	1,000K	$0.40	$2.40
Zhipu / Z.ai	GLM-5	`glm-5` (2026-02-11)	2026-02-11	200K	$1.00	$3.20
	GLM-4.7	`glm-4.7`(2025-12-23)	2025-12-23	200K	$0.60	$2.20
Moonshot	Kimi-K2.5	`kimi-k2.5` (2026-01-27)	2026-01-27	262K	$0.60	$3.00
MiniMax	MiniMax-M2.5	`MiniMax-M2.5` (2026-02-12)	2026-02-12	197K	$0.30	$2.40

answer instead. Abstention was available on both items and no model took it, so a failure to locate memory surfaces as a confident answer rather than an admission.

The Evaluate case isolates the disposition that distinguishes this level from the two below it. In Figure 15 the query names an appliance from a different manufacturer than the record establishes, and all 14 models answer the question as asked without ever noting that their own memory says otherwise. The failure is again not one of retrieval: the record sits in context and nothing about the query makes it hard to find. What is missing is the prior question of whether the retrieved fact should govern the request at hand—a question the models never raise, and therefore never answer. The interference subtype probes the same judgement from the opposite side, penalising a model that lets a stored preference govern a request it was never formed about; that both subtypes remain among the hardest in Table 2.3 indicates that deference to the query and deference to memory are not two ends of one dial that a model can simply be tuned along.

E.3 Cost Analysis

We use GPT-5.2 (`gpt-5.2-2025-12-11`) with reasoning enabled for the entire data construction pipeline. The pipeline covers six sub-tracks (understand-personalized, understand-experience, apply-user, apply-assistant, evaluate-interference, and evaluate-counterfactual), each going through memory generation, conflict checking, query/answer construction, and the subsequent multi-turn session construction (script, dialogue, conflict repair, style polish, and answer-turn extraction). For our 100-user benchmark with on average 38.4 queries per user, a single end-to-end run amounts to roughly 400 LLM calls per user, consuming on the order of 1.2 M input tokens and 1.0 M output tokens per user (reasoning tokens are billed as output). Under GPT-5.2's official pricing of $1.75 per 1 M input tokens and $14 per 1 M output tokens, this translates to about $16 per user for the full construction pipeline. Aggregated over all 100 users (approximately 120 M input tokens and 100 M output tokens in total), the end-to-end construction cost of the entire benchmark is approximately $1.6 K USD.

We estimate the total API expenditure of our evaluation using the per-token list prices reported in Table 8. All 14 models are evaluated on the same 3,840 queries under a 128K-token input, and the eight long-context models in the upper block of Table 8 (Qwen3.5-Plus, Qwen3.6-Plus, Gemini 2.5/3/3.1-Flash variants, GPT-5.4, Claude Sonnet 4.6 and Claude Opus 4.6) are additionally evaluated at a 1M-token input on the same queries.

CASE C1 in MUSE-Bench — USER-MEMORY UTILISATION

DIALOGUE HISTORY

[S3] *605 days before the query*

U I'm reordering kitchen gloves from a US store and I always get tripped up by their sizing. Back when I was buying kitchen gloves from that US store, **I used to measure my hand circumference as about 20.0 cm** around my knuckles (excluding the thumb), **which I converted to 7.87 inches**.

A If the site uses inches, keeping your converted number (7.87 in) on hand makes reordering much easier. . .

[S39] *369 days before the query*

U I'm ordering new kitchen cleaning gloves online for a deep-clean this weekend. Can you help me figure out the right size?

A I need your hand circumference around your knuckles (excluding the thumb), your hand length, and whether you prefer a snug, regular, or roomy fit.

U For kitchen gloves, **I measure my hand circumference at 18.5 cm** around my knuckles (excluding the thumb). I want them snug but not restrictive.

CURRENT QUERY

U I'm ordering replacement kitchen gloves from a US site that only lists sizes in inches — what hand circumference in inches should I use for my usual measurement?

CANDIDATE RESPONSES

A GENERIC SIZE
7.00 in

B CORRECT ✓
7.28 in (18.5/2.54)

C STALE VALUE
7.87 in (20.0/2.54, from [S3])

D MISCOMPUTED
6.28 in

E ABSTAIN
I don't know.

ANNOTATION

UTILISATION The user has stated this measurement twice, 605 and 369 days back, with different values. [S3] is explicitly retrospective ("I *used to* measure. . . ") and even records the converted figure; [S39] gives the measurement she uses now. **All 14 frontier LLMs answered C**, reproducing the string *7.87 in* verbatim from the older session. [S3] matches the query in surface form almost exactly — reordering kitchen gloves from a US store that lists inches — while [S39] matches it in substance. Note that nothing here requires computation the models cannot do: 18.5/2.54 is a single division, and no model chose the arithmetically botched option D. What fails is deciding which of two remembered values is still in force.

Figure 13: A representative USER-MEMORY UTILISATION instance on **MUSE-Bench**.

F RELATED WORK

F.1 MEMORY BENCHMARK

A growing body of work has proposed benchmarks (Jiayang et al.; Zhao et al., 2026) to evaluate memory capabilities of LLMs and LLM-based agents, which can be organized into the following categories.

Long-Term Conversational Memory. This line of work evaluates whether LLMs can recall and reason over information across extended multi-session dialogues. LoCoMo Maharana et al. (2024) is an early benchmark testing long-term memory through QA and summarization over multi-session histories. LongMemEval (Wu et al., 2025) systematizes evaluation into five core abilities including

CASE C2 in MUSE-Bench — ASSISTANT-MEMORY UTILISATION

DIALOGUE HISTORY

[S48] *295 days before the query*

U I'm flying through O'Hare on a short work trip and my layover is tight. I'm a sales rep, so I need reliable Wi-Fi and somewhere quiet enough to take a couple of calls. Which ORD lounges are actually worth targeting for a quick stop?

A Here's a short target list of 5 lounges at ORD worth aiming for on a quick work layover: 1) **United Club** (multiple locations); 2) **American Airlines Admirals Club**; 3) **Delta Sky Club** (Terminal 5); 4) **Swissport Lounge** (Terminal 5); 5) **The Club at ORD** (Terminal 1).

CURRENT QUERY

U My coworker just texted me the lounges we ended up using at ORD: **United Club, Delta Sky Club, and the British Airways Lounge**. Which of those are on the short list of lounges you recommended to me last year?

CANDIDATE RESPONSES

A INCOMPLETE
United Club

B QUERY ECHOED
United Club; Delta Sky Club; British Airways Lounge

C INCOMPLETE
Delta Sky Club

D CORRECT ✓
United Club; Delta Sky Club

E ABSTAIN
I don't know.

ANNOTATION

UTILISATION Answering requires intersecting the three lounges the coworker named with the five the assistant recommended 295 days earlier. The British Airways Lounge was never on that list, so a correct answer must drop one of the items the user just supplied. **7 of the 14 frontier LLMs answered B**, returning the coworker's three lounges unchanged. This is the answer one gives with no list in hand: nothing was recovered from memory, so the models fell back on the only set in front of them. Had they found the earlier recommendation and merely mis-computed the overlap, they would have landed on A or C, which together drew a single vote. Option E was available and no model took it, so the failed retrieval surfaced as a confident guess rather than an admission.

Figure 14: A representative ASSISTANT-MEMORY UTILISATION instance on **MUSE-Bench**.

cross-session reasoning, temporal reasoning, and knowledge updating. Subsequent work has expanded in scale and cognitive depth (Tavakoli et al., 2025; Shen et al., 2026a; Hu et al., 2026a).

Personalization and User Preference. A parallel line targets user modeling and preference following. PrefEval (Zhao et al., 2025a) evaluates implicit and explicit preference adherence; PersonaMem (Jiang et al., 2025a) and PersonaMem-v2 (Jiang et al., 2025b) assess response selection aligned with evolving user profiles; PersonaBench (Tan et al., 2025) and PersonaLens (Zhao et al., 2025b) benchmark personal information extraction and task-oriented personalization, respectively. KnowMe-Bench (Wu et al., 2026) pushes toward deep person understanding through autobiographical narratives. Recent work also examines failure modes: OP-Bench (Hu et al., 2026b) quantifies over-personalization, while RPEval (Feng et al., 2026) diagnoses irrational use of preference memory.

Agentic and Task-Oriented Memory. Beyond passive recall, several benchmarks evaluate whether agents can actively leverage memory for downstream actions. Mem2ActBench (Shen et al., 2026b) tests memory-driven tool selection and parameter grounding; MemoryArena (He et al., 2026) evaluates memory in interdependent multi-session agentic tasks; RealMem (Bian et al., 2026) situates

CASE C3 in MUSE-Bench — COUNTERFACTUAL

DIALOGUE HISTORY

[S90] *54 days before the query*

U My fridge's water flow has gotten noticeably slower, so I'm guessing the filter is overdue. Can you help me figure out the right replacement filter?

A To identify the correct replacement I just need one detail: what's your refrigerator make/model, or the part number printed on the current filter?

U **My refrigerator is a Whirlpool WRX735SDHZ.** Which filter fits that model, and where's the best place to buy it?

A For the Whirlpool WRX735SDHZ, the standard compatible filter is **EveryDrop Filter 2 (EDR2RXD1)**. Replacement interval: typically every 6 months…

CURRENT QUERY

U I'm ordering a replacement water filter for my 2026 maintenance checklist. **I remember my fridge is a GE GFE28GYNFS**, so which filter model should I buy?

CANDIDATE RESPONSES

A WRONG CORRECTION

Heads-up — I have your fridge down as a Frigidaire Gallery FGHB2868TF, not a GE GFE28GYNFS. The FGHB2868TF takes the Frigidaire PureSource Ultra II (EPTWFU01).

B GENERIC DEFAULT

The safest approach is to order by the part number printed on your current cartridge. If it has worn off, most units take a universally compatible "Filter 3" cartridge (EDR3RXD1).

C PREMISE FOLLOWED

For your GE GFE28GYNFS, the standard replacement is GE's XWFE filter, which supersedes the earlier RPWFE. Look for the XWFE marking when you reorder.

D CORRECT ✓

Heads-up — In my record, your refrigerator is Whirlpool WRX735SDHZ. It doesn't match what you said. Have you replaced it with a new refrigerator? If it's still the same Whirlpool model, use EDR2RXD1; if it's really GE, use XWFE.

E ABSTAIN

I don't know.

ANNOTATION

COUNTERFACTUAL The query names an appliance from an entirely different manufacturer than the one established in [S90]. **All 14 frontier LLMs chose C**: they answered the GE question competently and never mentioned that their own memory says otherwise. That the whole field lands in the same bucket is the point — this is not a retrieval failure, but unconditional deference to whatever the user asserts, with no check against what the model already knows. The expected behaviour does not even require assuming the user still owns the same fridge: had he replaced it, surfacing the discrepancy first, as D does, would still be the right move.

Figure 15: A representative COUNTERFACTUAL instance on **MUSE-Bench**.

evaluation in realistic project-oriented interactions. MemoryBench (Ai et al., 2025) uniquely emphasizes continual learning from user feedback rather than mere context comprehension.

Multimodal and Privacy-Aware Memory. Mem-gallery (Bei et al., 2026) extends evaluation to multimodal multi-session dialogues with interleaved image-text histories. EMemBench (Li et al., 2026b) uses interactive games to procedurally generate verifiable memory queries along agent trajectories. CIMemories (Mireshghallah et al.) evaluates contextual integrity, testing whether models appropriately disclose or withhold information across sessions.

F.2 RAG-BASED MEMORY TECHNIQUES

Retrieval-augmented generation (RAG) has become the dominant paradigm for equipping LLM agents with long-term memory, where historical interactions are stored in non-parametric stores and retrieved into the context on demand (Packer et al., 2023; Zhong et al., 2024; Chhikara et al., 2025; Rasmussen et al., 2025; Hu et al., 2025; Chen et al., 2025; Wang et al., 2024b; Li et al., 2026c)

The first line of work focuses on improving retrieval quality and memory organization (Kim et al., 2024; Li et al., 2024; Ong et al., 2025; Bae et al., 2022; Wang & Chen, 2025; Lu et al., 2023). Rather than relying on flat vector search, several methods introduce structured representations—such as hierarchical document trees (Sarthi et al., 2024), knowledge graphs with community summarization (Edge et al., 2024; Guo et al., 2024), or neurobiologically-inspired indexing that orchestrates knowledge graphs with personalized PageRank (Gutiérrez et al., 2024; 2025)—to capture complex inter-entity dependencies and support multi-hop reasoning. Another direction (Kirmayr et al., 2025; Xu et al., 2022; Zhang et al., 2024; Wang et al., 2025a; Li et al., 2026a) enhances the retrieval process itself: Self-RAG (Asai et al., 2023) enables adaptive, on-demand retrieval with self-reflection tokens; MemoRAG (Qian et al., 2025) builds a compressed global memory to generate retrieval clues before evidence lookup; and ComoRAG (Wang et al., 2026a) introduces iterative reasoning cycles with a dynamic memory workspace for stateful long-narrative comprehension.

The second line of work treats memory management itself as an agentic capability (Ho et al., 2025). Generative Agents (Park et al., 2023a) pioneered the memory stream architecture combining recency, relevance, and importance scoring with periodic reflection to synthesize higher-level abstractions. ReadAgent (Lee et al., 2024) compresses documents into gist memories and retrieves original passages on demand. Recent efforts further push toward learned memory control, where memory read/write/forget operations are optimized end-to-end via reinforcement learning (Bursa, 2026).

F.3 RL-BASED MEMORY TECHNIQUES

RL-Based Memory for LLM Agents. A growing line of work applies reinforcement learning to train LLMs to autonomously manage and utilize external memory, replacing static, heuristic-driven pipelines with learned memory policies (Yan et al., 2025; Zhou et al., 2025; Yu et al., 2026; Wang et al., 2026b; Liu et al., 2026b; Le et al., 2025). These methods typically expose memory operations (e.g., add, update, delete, retrieve) as actions in an RL environment and optimize the agent's memory policy via outcome-driven rewards using algorithms such as PPO or GRPO.

Among them, Memory-R1 (Yan et al., 2025) trains a Memory Manager and an Answer Agent with GRPO to learn structured memory operations and memory distillation for multi-session dialogue. MEM1 (Zhou et al., 2025) learns a compact internal state that jointly supports memory consolidation and reasoning, achieving constant memory footprint across long multi-turn tasks. AgeMem (Yu et al., 2026) unifies long-term and short-term memory management via a three-stage progressive RL strategy with step-wise GRPO. MemexRL (Wang et al., 2026b) introduces indexed experience memory with reward shaping to learn what to summarize, archive, index, and retrieve under a context budget. EMPO (Liu et al., 2026b) combines memory-augmented prompting with hybrid on- and off-policy optimization to improve exploration. Memory-R+(Le et al., 2025) leverages episodic memory for intrinsic motivation to enable RL fine-tuning of tiny LLMs. LearnToMemorize (Zhang et al., 2025b) formulates a memory cycle of retrieval, utilization, and storage, optimized via contrastive learning and DPO. A related direction trains LLMs to interact with external retrieval systems via RL (Jin et al., 2025), where models learn to generate search queries and reason over retrieved documents through outcome-based and process-based rewards.

G PROMPTS

Prompt 1: Memory Detection

TASK

You receive *one* user message (`query_block`), stamped with its absolute time and with the interval since the user's previous session, together with *all* of that user's earlier conversations (`history_block`), segmented into per-session blocks labelled `S1`, `S2`, ... in chronological order. Decide whether answering the query **requires memory carried by those blocks**, and identify the blocks that supply it.

DECISION CRITERIA

- **Content dependence.** The block holds material the answer needs facts, code, text, or an earlier request that the query continues.
- **Behavioural dependence.** The block governs how the answer must be delivered — the user's preferences, habits, language, or expected output format.
- **Counterfactual test.** Ask whether an assistant holding these blocks would answer better than one that had never seen them.
- **Evidence requirement.** Name every block whose memory is used.

OUTPUT SCHEMA (RETURN ONE JSON OBJECT ONLY; NO EXTRA TEXT)

```
{
  "uses_memory": <bool>,
  "ids": ["S3", ...],
  "quote": "<verbatim span, or null>",
  "reason": "<one short sentence, <= 15 words>"
}
```

FIELD CONSTRAINTS

- `uses_memory`: boolean.
- `ids`: identifiers of the blocks used, drawn from those shown; empty when `uses_memory` is `false`.
- `quote`: a verbatim span of 10–30 words taken from a named block.
- `reason`: a single short justification ($\leq$ 15 words).

INPUT

```
query_block: <current user message>
history_block: [S1] <interval>
               user: <message>
               assistant: <message>
```

Figure 16: The Prompt of Memory Detection.

Prompt 2: Memory Utilization Classification

TASK

You receive *one* user message (`query_block`), already verified to use conversation memory, together with *only* the session blocks that were identified as used (`history_block`). Classify **how** the query uses that memory into one of two labels: A or B.

LABEL DEFINITIONS

A – Memory Recall. The memory already contains the substance of the correct answer, so returning or lightly rephrasing it suffices for a good response, either verbatim, as when an earlier question is repeated or the query asks which item was meant and the prior conversation names it, or re-expressed by translating, condensing, or reformatting it. Re-expression takes effort but adds no information beyond what the memory already holds, and therefore remains A.

B – Memory Utilize. A correct answer must state something the memory does not contain: the model has to reason over what was said earlier—often across several turns of the prior conversation—or read off the preferences, rules, and precedents recorded there and apply them to the present query. It must also judge how each piece of memory bears on that query, drawing on the ones that genuinely inform the answer and setting aside those that are outdated, contradicted, or merely superficially similar, since carrying them over would degrade the answer. Here memory is an input to the answer rather than the answer itself.

OUTPUT SCHEMA (RETURN ONE JSON OBJECT ONLY; NO EXTRA TEXT)

```
{
  "answer_is": "<what a correct answer must contain, <= 12 words>",
  "purpose": "recall" | "utilize",
  "reason": "<one short sentence, <= 15 words>"
}
```

FIELD CONSTRAINTS

- `answer_is`: stated before the label, so that the label follows from it rather than from the surface form of the query ($\leq$ 12 words).
- `purpose`: exactly one of `"recall"` (A) or `"utilize"` (B), lower-case.
- `reason`: a single short justification ($\leq$ 15 words).

INPUT

```
query_block: <current user message>
history_block: <blocks in chronological order>
```

Figure 17: The Prompt of Memory Utilization Classification.

Prompt 3: Memory-Grounded Question Answering

TASK

You are an intelligent dialog bot. You will be shown `History Dialogs` between a user and an assistant, followed by a new `Question` from the same user on a specific date. Carefully read, memorize, and reason over the History Dialogs, then choose the single best option that answers the Question.

RULES

- This is a multiple-choice question. Exactly one option is correct.
- Base your answer on the information available in the `History Dialogs` and the `Question`. Do not invent facts that are not supported.
- If the History Dialogs do not contain enough information to answer confidently, pick the option that explicitly says you don't know (e.g. "I don't know.") if such an option is provided.
- Output **ONLY** the label of the chosen option (a single uppercase letter such as A, B, C, D or E). Do not output any other text, explanation, punctuation, or whitespace around the letter.

INPUT

```
History Dialogs:


{retrieved_texts}

Query Date: {query_date}

Question: {question}

Options:
{options}
```

OUTPUT SCHEMA

```
<single uppercase letter: A | B | C | D | E>
```

Figure 18: The Question Answering Prompt.

Prompt 4: Memory Cognitive-Demand Classification

TASK

You receive a user query and the historical memory that was confirmed to be used in answering it. Classify the **cognitive demand** of using this memory into one of three levels.

LEVEL DEFINITIONS

- **Understand.** The model needs to comprehend the requirements, preferences, or context recorded in the memory and apply them faithfully. The memory states what is needed—a preference, a style, a standing instruction, or relevant background—and the model honours it as written.
- **Apply.** The model needs to perform non-trivial reasoning or derivation before the memory can bear on the current query. For example, synthesising a constraint that no single memory entry states alone, inferring a pattern from past examples rather than an explicit rule, or carrying a correction to a case it was not originally about.
- **Evaluate.** Information in the memory or the query may interfere with producing a good answer. The model must judge on its own whether to trust, override, or set aside certain memory content—e.g. a previously stated preference that no longer fits, a premise in the query that the memory contradicts, or an outdated claim that would mislead the current response.

OUTPUT SCHEMA

```
{
  "level": "understand" | "apply" | "evaluate",
  "reason": "<one-sentence justification>"
}
```

INPUT

```
query_block: <current user message>
history_block: <memory blocks confirmed to be used>
```

Figure 19: The prompt for classifying the cognitive demand of memory utilization into three levels.