

RealMem: Benchmarking LLMs in Real-World Memory-Driven Interaction

Haonan Bian^{1,*}, Zhiyuan Yao^{2,*}, Sen Hu^{3,*†}, Zishan Xu⁴, Shaolei Zhang⁵,
Yifu Guo⁶, Ziliang Yang¹, Xueran Han⁵, Huacan Wang⁷, Ronghao Chen³

¹XDU ²ZJU ³PKU ⁴SJTU ⁵RUC ⁶SYSU ⁷UCAS

*Equal contribution †Corresponding author: husen@pku.edu.cn

Abstract

As Large Language Models (LLMs) evolve from static dialogue interfaces to autonomous general agents, effective memory is paramount to ensuring long-term consistency. However, existing benchmarks primarily focus on casual conversation or task-oriented dialogue, failing to capture “long-term project-oriented” interactions where agents must track evolving goals. To bridge this gap, we introduce **RealMem**, a benchmark explicitly designed to evaluate agent memory in long-term, project-oriented interaction paradigms. RealMem comprises over 2,000 cross-session dialogues across eleven scenarios, utilizing natural user queries for evaluation. We propose a synthesis pipeline that integrates Project Foundation Construction, Multi-Agent Dialogue Generation, and Memory and Schedule Management to simulate the dynamic evolution of memory. Experiments reveal that current memory systems face significant challenges in managing long-term project states and dynamic context dependencies inherent in sustained, project-oriented interactions. Our code and datasets are available at <https://github.com/AvatarMemory/RealMemBench>.

1 Introduction

Driven by the remarkable capabilities of recent Large Language Models (LLMs) (Guo et al., 2025; Comanici et al., 2025; Achiam et al., 2023), there is a growing consensus that AI agents must evolve from turn-based chatbots into long-term companions capable of sustained, context-aware collaboration (Tran et al., 2025; Sapkota et al., 2025). In this paradigm shift, memory systems have emerged as a foundational component (Hu et al., 2025). Robust memory is essential not only for practical applications such as personalized chatbots (Li et al., 2025a) and financial analysis (Zhang et al., 2024), but also as a key enabler toward Artificial General Intelligence (AGI) (Fang et al., 2025a).

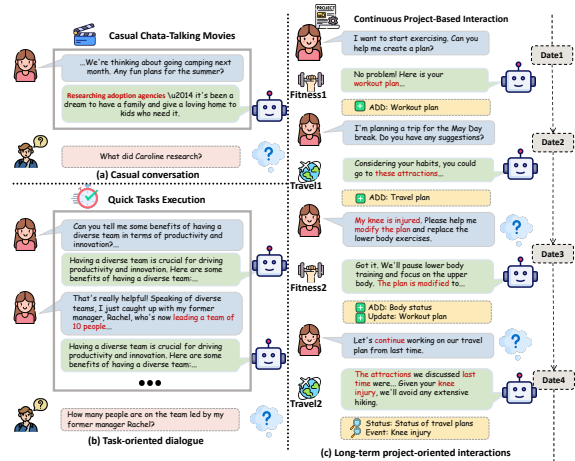


Figure 1: Comparison of three interaction paradigms in human-agent interactions: (a) casual conversation, (b) task-oriented dialogue, and (c) long-term project-oriented interactions spanning multiple sessions with interleaved projects and evolving context.

Effective long-term memory is essential for agents to maintain coherent collaboration across sustained, project-oriented interactions (Zhang et al., 2025b). We posit that long-term project-oriented interactions is governed by four fundamental imperatives: (1) Endogenous Query Nature: Queries arise organically from task progression, rather than appearing as isolated fact-checking questions. (2) Interleaved Distribution: User inquiries are interwoven across fragmented sessions. (3) Dynamic State Evolution: The interaction environment is inherently non-stationary, demanding continuous synchronization of memory with constantly evolving information states (Majumder et al., 2023; Zhang et al., 2025a). (4) Proactive Contextual Alignment: Agents should proactively resolve ambiguous intents by leveraging memory details (e.g., schedule management), and simultaneously maintain granular state updates that capture situational transitions rather than simplistic factual overwrites.

Benchmark	RealMem	HaluMem	LongMemEval	LoCoMo
Dialogue Type	Long-term project-oriented	Persona Creation	Task-oriented	Casual conversation
QA Timing	Interleaved within sessions	After each session	After all sessions	After all sessions
Query Source	Natural user queries	External queries	External queries	External queries
Proactive Alignment	✓	×	×	×
Project States Memory	✓	×	×	×
Max Memory Items	20	9	6	19
Memory Content	Project States, Schedules, Personas	Personas, Isolated Events, Relationships	Personas, Isolated Events	Personas, Isolated Events

Table 1: Comparison of representative memory benchmarks. RealMem distinguishes itself by introducing **proactive alignment** and **project state memory** into long-term interactions, shifting the evaluation paradigm from isolated fact-checking to continuous, **natural dialogue** progression.

To contextualize these requirements, we categorize user-agent interactions into three paradigms, as illustrated in Figure 1: *casual conversation*, *task-oriented dialogue*, and *long-term project-oriented interactions*. We argue that the third paradigm constitutes a crucial component of real-world memory-driven interactions. For example, in a fitness program spanning six months, an AI assistant can function as a personal trainer, providing continuous guidance by leveraging accumulated memories from past interactions. Existing dialogue memory benchmarks primarily focus on the first two interaction paradigms. For example, LoCoMo (Maharana et al., 2024) is limited to simulating human-human social chit-chat, while LongMemEval (Wu et al., 2024) approximates task dialogues via artificial “needle-in-a-haystack” tests. Consequently, their information flow remains discrete and episodic, failing to reflect the coherence of real-world workflows.

To bridge this gap, we introduce **RealMem**, a benchmark explicitly constructed to evaluate Real-World Memory-Driven Interaction through the lens of long-term projects. Grounded in eleven representative scenarios requiring robust long-term memory, RealMem comprises over 2,000 cross-session dialogues, thereby shifting the evaluation paradigm from isolated fact retrieval to project-centric memory utilization. RealMem evaluates agents using natural user queries that are organically interwoven across fragmented sessions. Our core focus is to assess its capacity to leverage accumulated memory and maintain the coherent thread

of a project to fulfill user requests within a realistic and evolving context.

We design a three-stage synthesis pipeline comprising *Project Foundation Construction*, *Multi-Agent Dialogue Generation*, and *Memory and Schedule Management*. This pipeline simulates the continuous evolution of long-term interactions, ensuring that memory is not predefined but dynamically emerges and evolves alongside the dialogue trajectory. Furthermore, we define a diverse set of query types (as illustrated in Figure 2) to rigorously evaluate the agent’s memory system. Through extensive evaluation on RealMem, we demonstrate that existing agent memory systems are inadequate for the dynamic requirements of long-term project-oriented interactions.

Our contributions are summarized as follows:

- We identify a critical gap between existing memory benchmarks and real-world memory-driven interaction: long-term project-oriented scenarios. To address this gap, we introduce **RealMem**, which shifts the evaluation focus from post hoc fact retrieval to the proactive use of memory within ongoing conversations to generate more effective responses.
- We propose a three-stage synthesis pipeline for constructing long-term project-oriented dialogues. This framework ensures global logical coherence across sessions and projects, while supporting the dynamic evolution of fine-grained memories.

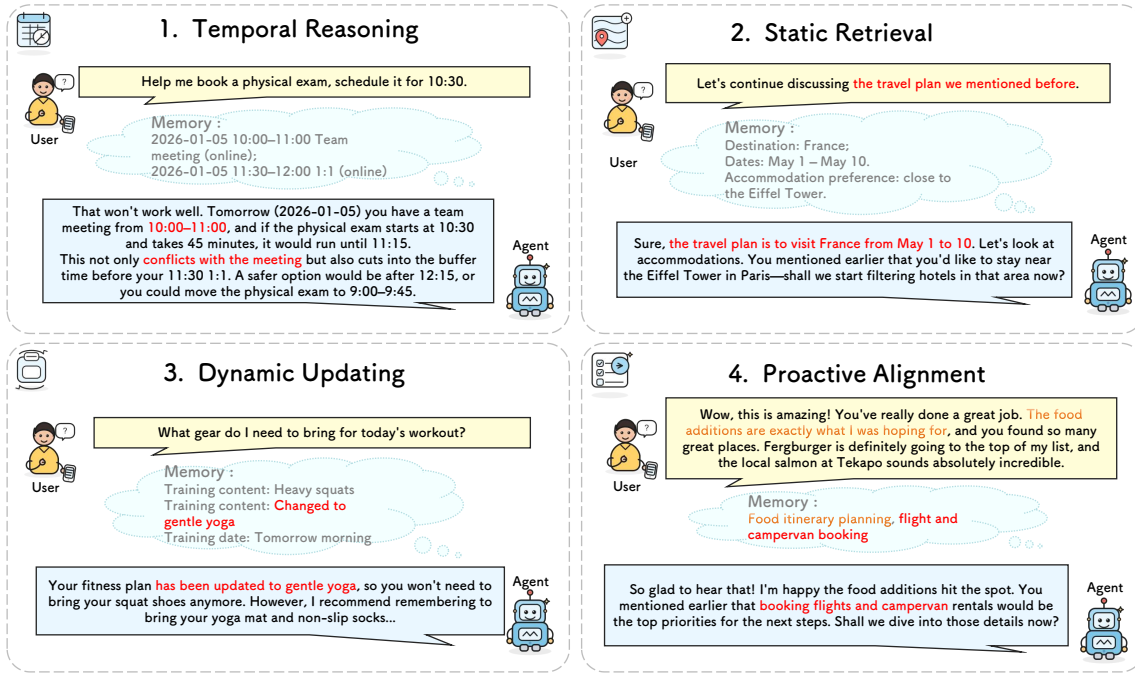


Figure 2: Examples of four query types in REALMEM: (1) **Temporal Reasoning** resolves temporal constraints and schedule conflicts; (2) **Static Retrieval** ensures continuity by recalling accumulated context; (3) **Dynamic Updating** synchronizes memory with evolving project states; and (4) **Proactive Alignment** leverages user memory to anticipate implicit intents and goals.

- Extensive evaluations reveal that state-of-the-art agent memory systems struggle to maintain **coherent project threads**. Our findings expose a critical gap in existing models’ ability to proactively align with evolving contexts in long-term interactions.

2 Related Work

2.1 Long-Term Memory Benchmarks

A number of benchmarks (Wei et al., 2025; Jiang et al., 2025) have been proposed to evaluate the memory capabilities of LLM-based agents. In this work, we focus on representative benchmarks most relevant to dialogue-based settings, as summarized in Table 1.

LoCoMo (Maharana et al., 2024) established an end-to-end evaluation paradigm for memory retention via question answering over ultra-long contexts, though it primarily assesses static information recall without explicit updates. LongMemEval (Wu et al., 2024) extended this framework by incorporating memory updating mechanisms to rigorously quantify retrieval precision and knowledge consistency across multi-year spans. HaluMem (Chen et al., 2025) utilizes injected persona profiles to assess hallucinations over extended timeframes, specifically measuring the consistency

of generated responses against pre-defined facts. However, these methods predominantly rely on externally constructed QA tasks to evaluate isolated factual details, often overlooking the dynamic state consistency essential for practical applications.

In contrast, RealMem targets long-term project-oriented interactions. It requires agents not only to recall static facts but to actively track evolving project states and handle interleaved queries that arise naturally across fragmented sessions.

2.2 Agent Memory Systems

To support effective long-term interaction, existing approaches commonly adopt RAG-based mechanisms for memory retrieval (Gutiérrez et al., 2025; Thang et al., 2025), and a variety of agent memory systems have been proposed. These systems typically distinguish between experiential memory (Tang et al., 2025b; Ouyang et al., 2025; Wang et al., 2024; Kim et al., 2025), derived from reflections on conversational history, and factual memory (Li et al., 2025b; Fang et al., 2025b; Wang and Chen, 2025; Rasmussen et al., 2025), which records knowledge acquired through agent–user and agent–environment interactions. Recent work has further explored memory-augmented reasoning in long-horizon settings, where agents leverage

structured visual or latent memory to support multi-step inference (Shi et al., 2026b,a), or distill intrinsic memory representations to improve computational efficiency (Hou et al., 2026). A-mem (Xu et al., 2025) proposes an agentic memory organization framework to improve memory effectiveness in long-term interactions. Mem0 (Chhikara et al., 2025) introduces a scalable memory-centric architecture with graph-based representations to capture relational structures across dialogue elements. MemoryOS (Kang et al., 2025) adopts operating-system-inspired memory management to improve retrieval and update efficiency. Graph Memory (Hu et al., 2026) further enhances the knowledge graph’s capacity to represent memory by allowing entity descriptions to record relevant events and remain up to date.

3 Method

To evaluate memory systems in complex and evolving contexts, we introduce the **RealMem** benchmark. Inspired by recent advances in agentic frameworks (Figure 3) that leverage multi-agent collaboration for complex problem solving (Huang et al., 2024; Liu et al., 2024; Tang et al., 2025a), RealMem simulates realistic user behavior in which multiple long-term projects are managed concurrently within a unified “SuperApp” environment. Its design is structured around three tightly coupled stages—static project foundation construction, dynamic interaction generation, and closed-loop memory feedback—which together enable systematic assessment of an agent’s ability to utilize, update, and maintain dynamic memory over extended interactions.

3.1 Project Foundation Construction

The **Project Foundation Construction** phase initializes the static context and structural scaffolding required for subsequent dialogue generation. As long-horizon generation demands explicit planning to preserve global coherence (Xia et al., 2025), we adopt a hierarchical strategy to construct a *Project Skeleton*, following a “blueprint-first” paradigm that has proven effective in recent multi-turn data generation frameworks (Prabhakar et al., 2025; Chen et al., 2025).

Specifically, this phase establishes a user *Persona* to encode demographics for behavioral consistency, and a *Project Goal* to specify a quantitative objective (e.g., “losing 15 kg in six months”)

as a clear long-term target. Crucially, we introduce *Project Attributes* as dynamic state variables that serve as the core mechanism for project modeling, tracking temporal evolution to capture changes in user progress and contextual state.

To reflect realistic user needs and assess the generality of our framework, we curate eleven representative scenarios spanning four application domains. For each scenario, we identify a set of core attributes—ranging from schedule adjustments to emotional regulation—to capture the multifaceted nature of long-term human–AI interaction.

The generation process proceeds in three stages. First, a *Project Blueprint* is constructed to outline high-level milestones. Second, an *Event List* is generated to encode causal dependencies among milestones. Third, *Session Summaries* are produced to guide individual dialogue sessions. This hierarchical decomposition ensures that local interactions remain aligned with the global project narrative, mitigating fragmentation commonly observed in long-context generation (Xia et al., 2025; Prabhakar et al., 2025). Finally, to simulate concurrent task management, session summaries from multiple projects are aggregated and interleaved to form a unified session queue.

3.2 Multi-Agent Dialogue Generation

Inspired by recent advances in multi-agent data synthesis (Liu et al., 2024; Xia et al., 2025), we adopt a dual-agent framework consisting of a *User Agent* and an *Assistant Agent* to simulate realistic user–agent interactions. Compared to *one-shot generation of multi-turn dialogues*, this simulation-based paradigm provides finer-grained control over interaction dynamics and policy constraints, as demonstrated in prior work such as IntellAgent (Levi and Kadar, 2025).

To ensure long-term consistency, each dialogue session is generated under a structured context spanning three dimensions: (1) static project background, (2) memory points extracted from historical interactions, and (3) established schedules. Dialogue generation proceeds at the session level, with sessions sequentially drawn from an interleaved queue that simulates concurrent project management.

For the User Agent, the context includes information about the current event together with a summary of the immediately preceding event, enabling coherent progression awareness without exposing future plans. Importantly, we restrict the

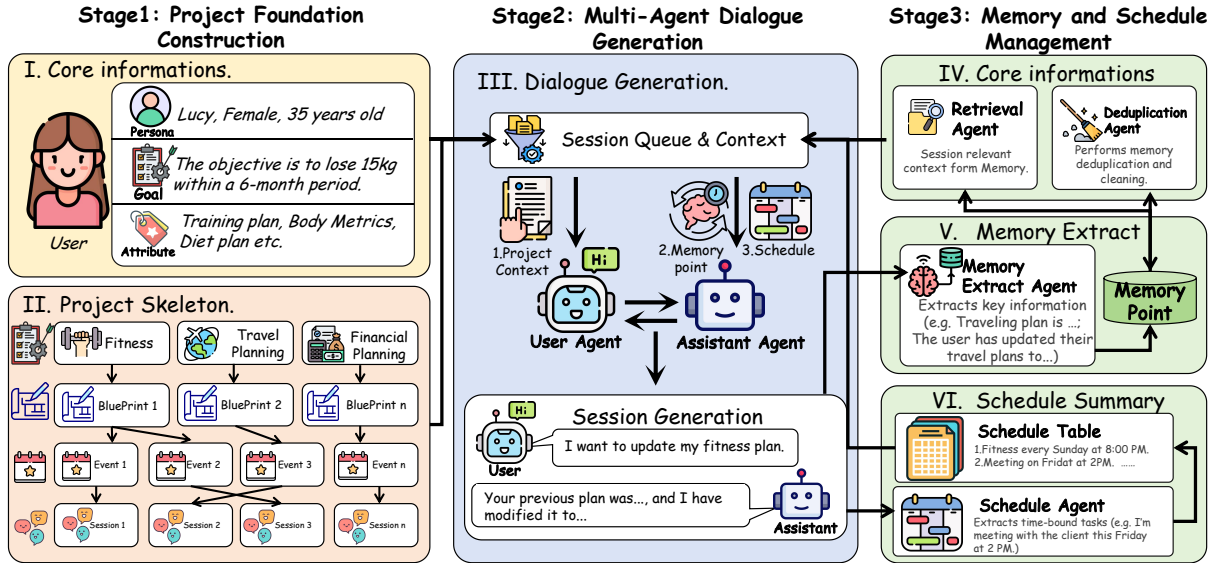


Figure 3: Overview of the data synthesis framework. The pipeline consists of three cascaded stages: (1) **Project Foundation Construction**, which initializes user personas and hierarchical project skeletons (i.e., blueprints, events, and sessions); (2) **Multi-Agent Dialogue Generation**, where the User Agent and Assistant Agent simulate interactions based on the session queue and dynamic context; and (3) **Memory and Schedule Management**, which iteratively retrieves, updates, and deduplicates memory points and schedule tables to ensure long-term consistency.

User Agent’s input to session-relevant summaries only, thereby enforcing explicit *task boundaries* and preventing premature reasoning over future tasks. In contrast, the Assistant Agent is provided with all memory points relevant to the current session to support informed decision making. To further mitigate temporal inconsistencies frequently observed in LLM-based generation (e.g., scheduling conflicts across projects), we incorporate a *Global Schedule* into the Assistant Agent’s context. This design substantially improves temporal alignment and logical coherence across long-term, multi-project interactions.

3.3 Memory and Schedule Management

This stage processes the generated dialogue data to form a closed feedback loop. To this end, our framework employs a set of specialized agents to validate, consolidate, and update the system state, ensuring that memory evolves accurately and consistently over time.

Specifically, a *Memory Extraction Agent* parses raw dialogues to identify salient facts and converts them into structured memory points. In parallel, a *Schedule Agent* detects tasks with explicit temporal attributes (e.g., scheduled meetings or deadlines) and updates the global schedule accordingly. To improve storage efficiency and reduce noise, we introduce a *Deduplication Agent* that per-

forms semantic-level deduplication over the memory base, removing redundant or overlapping entries.

In subsequent dialogue generation cycles, the *Session Summary* is used as a query to retrieve relevant information from the optimized memory base, which is then provided as contextual input to the Assistant Agent. This closed-loop design enables continuous memory refinement and supports coherent long-term interactions across sessions.

4 Evaluation

To comprehensively assess retrieval quality, we adopt a hybrid evaluation strategy that combines standard quantitative metrics with LLM-based semantic judgments (Liu et al., 2023).

For retrieval performance, we report **Recall@k** and **NDCG@k** ($k \in \{10, 20\}$) to measure strict ranking accuracy. To capture semantic relevance beyond exact lexical matching, we further introduce LLM-based metrics: **Mem Recall**, which assesses the semantic coverage of relevant information, and **Mem Helpful**, which evaluates the practical usefulness of the retrieved context for answering the query.

For generation quality, we employ the **QA Score**, which is derived from a response consistency rubric (ranging from conflicting to fully state-aligned). This metric explicitly evaluates

whether the agent correctly incorporates the user’s dynamic state, rather than merely producing fluent but context-agnostic responses.

5 Experiment

5.1 Experiment Setup

We conduct a systematic evaluation of representative state-of-the-art memory systems on our proposed RealMem benchmark, including Mem0 (Chhikara et al., 2025), A-mem (Xu et al., 2025), MemoryOS (Kang et al., 2025), and Graph Memory (Hu et al., 2026). For A-mem, MemoryOS, and Graph Memory, we strictly follow the embedding models and hyperparameter settings recommended in their original papers; for Mem0, we use text-embedding-3-small (OpenAI, 2024) as its default embedding model. We utilize GPT-4o-mini for memory extraction. For the answer generation phase, we employ both GPT-4o-mini and GPT-4o (Hurst et al., 2024) to assess performance across different backbones. Finally, to ensure consistency, we adopt GPT-4o as the LLM-as-a-Judge for all evaluations.

During the answer generation stage, we evaluate each method under two independent context construction settings. In the memory-only setting (Value = memory), the model is provided solely with the Top-20 memory entries retrieved by the memory system. In the session-based setting (Value = session), memory entries are not directly used; instead, the model receives the corresponding Top-5 original dialogue sessions associated with the retrieval results, in order to assess generation performance based on session-level context.

We note that MemoryOS internally maintains memory at the page level and does not support alignment with original dialogue sessions; as a result, it cannot be evaluated under the session-based setting.

5.2 Overall Evaluation

The results on generation tasks offer critical insights into the architectural requirements for long-term project agents. First, under the **Memory-only** setting, the superior performance of **MemoryOS** validates the effectiveness of hierarchical memory architectures (e.g., STM, MTM, and LTM). It demonstrates that effective compression and indexing of key information allow agents to maintain high accuracy even without full session history. Second, when **Session** context is available, **Graph**

Memory achieves the highest scores. This indicates that the **RealMem** benchmark involves intricate entity relationships, where graph structures excel at capturing complex dependencies, thereby providing more precise context than linear history alone.

Furthermore, a cross-model comparison reveals the challenging nature of realistic project scenarios. The QA accuracy of **GPT-4o** is consistently higher than that of **GPT-4o-mini**. Crucially, even with Oracle memory, GPT-4o-mini exhibits limited performance. This suggests that solving the complex queries in RealMem necessitates not only accurate retrieval but also high-performance foundation models with strong reasoning capabilities.

Additionally, the **Full Context** baselines reveal that despite access to complete dialogue history, **Full Context (GPT-4.1-Mini)** achieves only 0.5156 on the Session-level QA-4o-Mini metric, still below the best retrieval-based methods. This suggests that extending the context window alone is insufficient, and structured memory retrieval remains essential for long-term project-oriented interactions. **LightMem** achieves competitive performance comparable to Mem0 and A-mem, yet still falls notably short of Oracle, consistent with the overall trend.

Finally, the substantial gap between all methods and the **Oracle** (e.g., Oracle Recall reaches 0.993) highlights significant room for improvement in capturing long-term context dependencies. This underscores a key finding: memory can effectively replace raw session history only when retrieval quality is sufficiently close to the theoretical upper bound.

The retrieval results further clarify the correlation between retrieval metrics and downstream generation quality in the RealMem benchmark. While **A-mem** achieves the highest Recall@20 (0.7235), its lower NDCG scores suggest it retrieves a broader but noisier set of information. Conversely, **Graph Memory** dominates in NDCG metrics (0.5654 @10), aligning with its superior generation performance in Table 3. This discrepancy highlights a critical characteristic of the benchmark: for long-term project dialogues, **precision and ranking quality** (NDCG) are more decisive than mere coverage (Recall). High recall with low precision introduces noise that distracts the LLM, whereas high NDCG ensures that the most relevant context is prioritized, directly translating to better response quality. See Appendix A.2

Method	Value = Memory				Value = Session			
	QA-4o-Mini	QA-4o	Mem Rec.	Mem Help.	QA-4o-Mini	QA-4o	Mem Rec.	Mem Help.
Full Context (GPT-4o-Mini)	–	–	–	–	0.411	–	–	–
Full Context (GPT-4.1-Mini)	–	–	–	–	0.516	–	–	–
Graph Mem	0.474	0.497	0.490	0.598	0.539	0.567	0.608	0.708
Mem0	0.449	0.514	0.529	0.634	0.526	0.609	0.594	0.666
A-mem	0.416	0.492	0.455	0.513	0.504	0.606	0.590	0.655
MemoryOS	0.490	0.567	0.532	0.606	–	–	–	–
LightMem	0.441	0.512	0.490	0.598	0.521	0.623	0.585	0.654
Oracle	0.683	0.804	0.993	0.922	0.580	0.696	0.767	0.845

Table 2: Generation performance comparison. We report **QA Score** (on **GPT-4o-mini** and **GPT-4o**), **Mem Recall**, and **Mem Helpful** scores. “Memory” denotes memory-only context, while “Session” includes full session context. **Bold** indicates the best performance among non-oracle methods. Note that Value = Session metrics are not applicable to MemoryOS due to its inherent mechanism, and Value = Memory metrics are not applicable to Full Context baselines as they do not employ a memory retrieval stage.

Method	R@10	R@20	NDCG@10	NDCG@20
Graph Mem	0.6344	0.7017	0.5654	0.5826
Mem0	0.6055	0.6515	0.5251	0.5398
A-mem	0.5760	0.7235	0.5211	0.5560

Table 3: Retrieval performance evaluated on session-level context. Metrics include Recall@K (R@K) and NDCG@K. **Bold** indicates the best performance.

for supplementary experiments.

5.3 Performance on Different Question Types

Table 4 presents a fine-grained performance analysis across different memory categories and session types. We summarize the key observations as follows:

Our results highlight distinct architectural strengths across different complexity dimensions. **MemoryOS** demonstrates robust superiority in handling dynamic information, achieving the highest scores in both *Static Retrieval* (QA 0.478) and *Dynamic Updating* (QA 0.521) categories. This validates the effectiveness of its hierarchical architecture in tracking evolving user states and managing fragmented updates. Conversely, **Graph Memory** dominates the *Temporal Reasoning* category (QA 0.375, Recall 0.559). It is worth noting that temporal reasoning queries account for only 1.7% of the total evaluation set, reflecting the natural frequency of complex temporal tasks in realistic project-oriented interactions rather than a limitation of coverage — these cases are retained under strict filtering criteria that exclude trivial date-retrieval questions and retain only queries requiring genuine planning, conflict detection, or complex temporal dependency reasoning. We plan

to explore specialized benchmarks with higher temporal reasoning query density in future work.

Regarding session types, performance across all models generally declines in *Multi Session* settings compared to *Single Session*, highlighting the inherent difficulty of maintaining long-range context. Nevertheless, **MemoryOS** maintains leading QA performance in both settings (Single: 0.505, Multi: 0.468), demonstrating its robustness in handling diverse interaction spans. Additionally, in the *Proactive Alignment* category, MemoryOS achieves the highest QA score (0.571), indicating a superior capability in anticipating future steps and aligning with the user’s long-term objectives.

5.4 Scenario-Based Performance Analysis

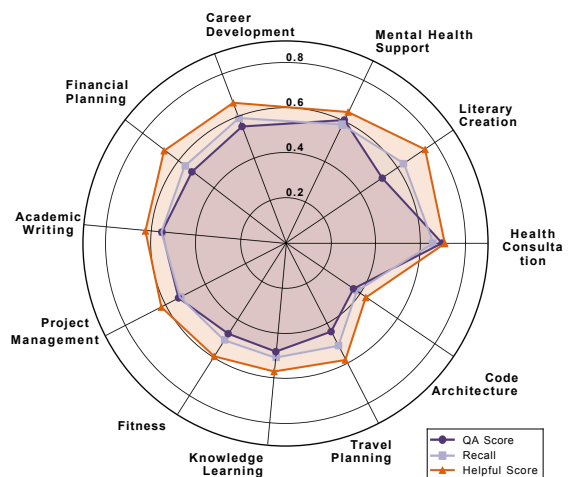


Figure 4: Average performance scores of MemoryOS across various topics.

In this section, we present the memory system performance across various topics. To visualize trends, we select *MemoryOS*, the most robust

Method	Memory Category								Session Type			
	Stat. Ret.		Dyn. Upd.		Proac. Align.		Temp. Reas.		Single Sess.		Multi Sess.	
	QA	Rec.	QA	Rec.	QA	Rec.	QA	Rec.	QA	Rec.	QA	Rec.
A-mem	0.404	0.465	0.453	0.427	0.483	0.436	0.278	0.361	0.421	0.454	0.409	0.457
Mem0	0.436	0.529	0.462	0.553	0.538	0.506	0.347	0.539	0.457	0.542	0.437	0.509
MemoryOS	0.478	0.536	0.521	0.557	0.571	0.490	0.319	0.448	0.505	0.565	0.468	0.482
Graph Mem	0.470	0.500	0.483	0.512	0.513	0.392	0.375	0.559	0.483	0.516	0.461	0.451

Table 4: Overall performance comparison (QA Score and Recall). Methods are listed in rows, while memory categories and session types are arranged in columns. Abbreviations used in the header: **Stat. Ret.** (Static Retrieval), **Dyn. Upd.** (Dynamic Updating), **Proac. Align.** (Proactive Alignment), and **Temp. Reas.** (Temporal Reasoning).

model, as the representative. Notably, other baselines exhibit similar tendencies; detailed results are provided in the Appendix.

As illustrated in Figure 4, the efficacy of MemoryOS reveals significant domain-dependent variations. The system exhibits its strongest performance in consultative and creative domains, such as *Mental Health Support*, *Health Consultation*, and *Literary Creation*. In these dimensions, the *Helpful Score* (indicated by the orange trajectory) consistently outperforms other metrics, approaching or exceeding the 0.6 to 0.8 range. This trend suggests that for open-ended, human-centric tasks, the system is capable of providing highly valuable feedback and maintaining a coherent narrative flow, even when precise *Recall* is not optimal.

Conversely, the system encounters severe challenges in highly technical and rigid domains. This is most distinct in *Code Architecture*, where all three metrics—*QA Score*, *Recall*, and *Helpful Score*—show a sharp contraction towards the center of the chart, dropping significantly below 0.4. Unlike soft-constraint tasks like *Financial Planning*, the rigorous dependency tracking and strict logical consistency required for complex engineering tasks expose a structural limitation in the current architecture’s ability to handle hard constraints effectively.

5.5 Efficiency and Cost Analysis of Long-Term Memory Systems

Table 5 reveals a critical bottleneck: memory incorporation latency consistently exceeds retrieval speeds across all systems. This underscores the urgent need to optimize memory ingestion mechanisms for scalable deployment.

Our evaluation of system latency and token consumption reveals a significant trade-off between operational overhead and model performance. Specif-

Method	Avg. Add Memory Time (s)	Avg. Retrieve Memory Time (s)	Cost Token
Graph Mem	26.898	0.18803	1224.02
Mem0	11.751	1.261	4178.18
A-mem	8.663	0.013	500.32
MemoryOS	14.755	0.151	9201.01

Table 5: Comparison of efficiency and cost across different memory systems. We report the average latency for memory addition and retrieval, alongside total token consumption.

ically, while MemoryOS achieves competitive retrieval latency (0.151 s), it incurs the highest token cost (9,201), indicating that its superior performance necessitates extensive context maintenance. Conversely, A-mem demonstrates the lowest latency and resource consumption, yet as noted previously, this efficiency comes at the expense of retrieval accuracy and response quality. Graph Memory offers a viable compromise by delivering retrieval speeds comparable to MemoryOS (0.188 s vs. 0.151 s) with significantly reduced token costs, although it requires longer processing times for memory addition.

5.6 Cross-Validation by Humans

System	Avg. QA Score	QA Rank	Avg. Human Rank
MemoryOS	1.40	1	1.60
Graph Mem	1.10	2	2.20
A-mem	1.00	3	2.60
Mem0	0.80	4	3.60

Table 6: Comparison of automated QA scores and human evaluation. The results demonstrate a strong alignment between our QA metrics and human judgments, with **MemoryOS** ranking first in both settings.

Table 6 presents the comparative results of average QA scores and human rankings. MemoryOS demonstrates superior performance, achieving the

highest QA score (1.40) and the best average human ranking (1.60). Crucially, the ranking order remains consistent across all evaluated systems (MemoryOS > Graph Memory > A-mem > Mem0), where a decrease in QA score strictly corresponds to a decline in human ranking (e.g., Mem0 scores lowest at 0.80 and ranks last at 3.60). This perfect alignment in system-level ranking further validates that our automated QA Score serves as a reliable indicator of model performance consistent with human preference. Full protocol details are provided in Appendix A.3.

6 Conclusion

We introduce **RealMem**, a benchmark for evaluating long-term memory in realistic, project-oriented interactions, where agents must track evolving goals and maintain coherent project states across sessions. RealMem departs from prior benchmarks by emphasizing sustained, multi-session projects and by providing a synthesis pipeline that enables controlled simulation of dynamic memory evolution. Experiments on RealMem show that existing memory systems remain fundamentally limited in handling long-term project dependencies. The substantial gap between all evaluated methods and the Oracle upper bound further highlights that effective long-term memory integration remains a core bottleneck for project-oriented agents. We believe RealMem provides a valuable diagnostic testbed for studying long-term agent memory and hope it will facilitate future research toward more robust, scalable, and reliable memory systems for real-world autonomous agents.

Limitations

Our data construction process relies significantly on the Gemini 2.5 series models for data collection, alongside human annotation for label verification. While this reliance may raise concerns regarding reproducibility and associated costs, the Gemini 2.5 models demonstrate a superior capability to simulate realistic human–computer interactions with highly controllable memory utilization. We note that our “real-world” framing refers specifically to the long-term, project-oriented interaction paradigm we target, rather than the data source itself; the synthesis pipeline was iteratively refined by referencing real human–AI conversations to ensure distributional fidelity. In terms of evaluation scope, temporal reasoning queries currently ac-

count for a small proportion (1.7%), reflecting their natural frequency under strict quality filtering, and we plan to explore higher-density temporal benchmarks in future work. Additionally, RealMem does not yet assess tool use capabilities, which we plan to incorporate to further extend its applicability.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Ding Chen, Simin Niu, Kehang Li, Peng Liu, Xiangping Zheng, Bo Tang, Xinchu Li, Feiyu Xiong, and Zhiyu Li. 2025. Halumem: Evaluating hallucinations in memory systems of agents. *arXiv preprint arXiv:2511.03506*.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.
- Gheorghe Comanici, Eric Bieber, Mike Schaeckermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.
- Jinyuan Fang, Yanwen Peng, Xi Zhang, Yingxu Wang, Xinhao Yi, Guibin Zhang, Yi Xu, Bin Wu, Siwei Liu, Zihao Li, and 1 others. 2025a. A comprehensive survey of self-evolving ai agents: A new paradigm bridging foundation models and lifelong agentic systems. *arXiv preprint arXiv:2508.07407*.
- Jizhan Fang, Xinle Deng, Haoming Xu, Ziyang Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, and 1 others. 2025b. Lightmem: Lightweight and efficient memory-augmented generation. *arXiv preprint arXiv:2510.18866*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*.
- Yubo Hou, Zhisheng Chen, Tao Wan, and Zengchang Qin. 2026. Flashmem: Distilling intrinsic latent memory via computation reuse. *Preprint, arXiv:2601.05505*.

- Sen Hu, Yuxiang Wei, Jiabin Ran, Zhiyuan Yao, and Lei Zou. 2026. [Does memory need graphs? a unified framework and empirical analysis for long-term dialog memory](#). *Preprint*, arXiv:2601.01280.
- Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, and 1 others. 2025. Memory in the age of ai agents. *arXiv preprint arXiv:2512.13564*.
- D Huang, JM Zhang, M Luck, Q Bu, Y Qing, and H Cui. 2024. Agentcoder: Multi-agent code generation with effective testing and self-optimization. *University of Hong Kong, King's College London, University of Sussex, Shanghai Jiao Tong University*.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Bowen Jiang, Yuan Yuan, Maohao Shen, Zhuoqun Hao, Zhangchen Xu, Zichen Chen, Ziyi Liu, Anvesh Rao Vijjini, Jiashu He, Hanchao Yu, and 1 others. 2025. Personamem-v2: Towards personalized intelligence via learning implicit user personas and agentic memory. *arXiv preprint arXiv:2512.06688*.
- Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. 2025. [Memory os of ai agent](#). *Preprint*, arXiv:2506.06326.
- Namyong Kim, Kai Tzu iunn Ong, Yeonjun Hwang, Minseok Kang, Iiseo Jihn, Gayoung Kim, Minju Kim, and Jinyoung Yeo. 2025. [Principles: Synthetic strategy memory for proactive dialogue agents](#). *Preprint*, arXiv:2509.17459.
- Elad Levi and Ilan Kadar. 2025. Intelligent: A multi-agent framework for evaluating conversational ai systems. *arXiv preprint arXiv:2501.11067*.
- Hao Li, Chenghao Yang, An Zhang, Yang Deng, Xiang Wang, and Tat-Seng Chua. 2025a. Hello again! llm-powered personalized agent for long-term dialogue. 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)*, pages 5259–5276.
- Zhiyu Li, Shichao Song, Chenyang Xi, Hanyu Wang, Chen Tang, Simin Niu, Ding Chen, Jiawei Yang, Chunyu Li, Qingchen Yu, and 1 others. 2025b. Memos: A memory os for ai system. *arXiv preprint arXiv:2507.03724*.
- Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, and 1 others. 2024. Toolace: Winning the points of llm function calling. *arXiv preprint arXiv:2409.00920*.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruo Chen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.
- Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. *arXiv preprint arXiv:2402.17753*.
- Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. 2023. Clin: A continually learning language agent for rapid task adaptation and generalization. *arXiv preprint arXiv:2310.10134*.
- OpenAI. 2024. [text-embedding-3-small \(model documentation\)](#).
- Siru Ouyang, Jun Yan, I Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T Le, Samira Daruki, Xiangru Tang, and 1 others. 2025. Reasoningbank: Scaling agent self-evolving with reasoning memory. *arXiv preprint arXiv:2509.25140*.
- Akshara Prabhakar, Zuxin Liu, Ming Zhu, Jianguo Zhang, Tulika Awalgaoankar, Shiyu Wang, Zhiwei Liu, Haolin Chen, Thai Hoang, Juan Carlos Niebles, and 1 others. 2025. Apigen-mt: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. *arXiv preprint arXiv:2504.03601*.
- Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025. Zep: a temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*.
- Ranjan Sapkota, Konstantinos I Roumeliotis, and Manoj Karkee. 2025. Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenges. *arXiv preprint arXiv:2505.10468*.
- Yaorui Shi, Yuxin Chen, Siyuan Wang, Sihang Li, Hengxing Cai, Qi Gu, Xiang Wang, and An Zhang. 2026a. [Look back to reason forward: Revisitable memory for long-context llm agents](#). *Preprint*, arXiv:2509.23040.
- Yaorui Shi, Shugui Liu, Yu Yang, Wenyu Mao, Yuxin Chen, Qi GU, Hui Su, Xunliang Cai, Xiang Wang, and An Zhang. 2026b. [Memocr: Layout-aware visual memory for efficient long-horizon reasoning](#). *Preprint*, arXiv:2601.21468.
- Shuo Tang, Xianghe Pang, Zexi Liu, Bohan Tang, Rui Ye, Tian Jin, Xiaowen Dong, Yanfeng Wang, and Siheng Chen. 2025a. Synthesizing post-training data for llms through multi-agent simulation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23306–23335.

- Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, and 1 others. 2025b. Agent kb: Leveraging cross-domain experience for agentic problem solving. *arXiv preprint arXiv:2507.06229*.
- Nguyen Thang, Peter Chin, Thang Nguyen, Yu-Wing Tai, Peter Chin, and Yu-Wing Tai. 2025. [Ma-rag: Multi-agent retrieval-augmented generation via collaborative chain-of-thought reasoning](#). <http://arxiv.org/abs/2505.20096>, abs/2505.20096.
- Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O’Sullivan, and Hoang D Nguyen. 2025. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*.
- Yancheng Wang, Ziyang Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. 2024. Recmind: Large language model powered agent for recommendation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4351–4364.
- Yu Wang and Xi Chen. 2025. [Mirix: Multi-agent memory system for llm-based agents](#). *Preprint*, arXiv:2507.07957.
- Tianxin Wei, Noveen Sachdeva, Benjamin Coleman, Zhankui He, Yuanchen Bei, Xuying Ning, Mengting Ai, Yunzhe Li, Jingrui He, Ed H Chi, and 1 others. 2025. Evo-memory: Benchmarking llm agent test-time learning with self-evolving memory. *arXiv preprint arXiv:2511.20857*.
- Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2024. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*.
- Haotian Xia, Hao Peng, Yunjia Qi, Xiaozhi Wang, Bin Xu, Lei Hou, and Juanzi Li. 2025. Storywriter: A multi-agent framework for long story generation. *arXiv preprint arXiv:2506.16445*.
- Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*.
- Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, and 1 others. 2025a. Agentic context engineering: Evolving contexts for self-improving language models. *arXiv preprint arXiv:2510.04618*.
- Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiase Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, and 1 others. 2024. A multi-modal foundation agent for financial trading: Tool-augmented, diversified, and generalist. In *Proceedings of the 30th acm sigkdd conference on knowledge discovery and data mining*, pages 4314–4325.
- Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2025b. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6):1–47.

A Appendix

A.1 Supplementary Details of RealMem

This appendix provides additional statistical information and key definitions of the RealMem dataset to support a more detailed understanding of its data composition and task taxonomy.

Statistic	Value
<i>Dialogue Scale</i>	
Avg. Context Length (tokens/user)	269,190
Avg. Session Num. (per user)	205
Avg. Turns per Session	6.8
Total Dialogue Turns	14,028
<i>Memory Composition</i>	
Total Memories	5,072
- Project-centric Event	4,135
- Persona	850
- Schedule	87
Used Memory (Ground Truth)	1,834
<i>Question Distribution</i>	
Total Questions	1,415
- Static Retrieval	1,075
- Dynamic Updating	156
- Implicit Preference	160
- Temporal-reasoning	24

Table 7: Statistical Overview of RealMem Datasets

A.1.1 Definition of Question Types

To systematically evaluate the capabilities of our memory mechanism, we categorize user queries into four distinct types based on their retrieval requirements and interaction logic:

- **Static Retrieval:** Queries that seek to advance the project state linearly based on confirmed context. In these scenarios, the user accepts the current progress (e.g., "The suggestions are great") and explicitly requests the subsequent step (e.g., "What do we need to do next?"). The agent must retrieve the latest stable state to build upon it without altering previous decisions.
- **Dynamic Updating:** Queries involving the modification, deletion, or conflict resolution of existing plans. Unlike incremental queries, the user introduces new constraints (e.g., "maintain 12 days but add West Coast") that conflict with established memories. The agent

must retrieve relevant constraints and perform trade-off analysis to generate a revised state.

- **Proactive Alignment:** Queries lacking explicit instructions, often manifested as emotional feedback or vague statements (e.g., "This is fantastic!"). Here, the agent cannot rely on current turn commands but must retrieve long-term user priorities or historical preferences to proactively propose the next logical action, transforming latent intent into execution.
- **Temporal-reasoning:** Queries requiring the processing of time-sensitive information and scheduling logic. These tasks involve validating proposed slots against existing commitments (e.g., checking for overlaps between a new workout plan and existing study sessions) or sequencing events chronologically. The agent must retrieve structured schedule data to perform constraint satisfaction checks.

A.1.2 Definition of Memory Types

To support effective retrieval and distinct reasoning tasks, we structure the agent’s memory into three specific categories:

- **Persona:** This category stores static and semi-static user attributes, including personal profiles, long-term goals, and specific preferences (e.g., dietary restrictions, preferred writing styles). It serves as the foundation for personalization, ensuring that the agent’s responses consistently align with the user’s identity and historical habits across different sessions.
- **Project States:** This category records the evolving state of the specific task or project. It encapsulates the core content generated during the collaboration, such as finalized itinerary details, code architecture decisions, or plot outlines. These memories act as the "knowledge base" for the project, allowing the agent to track progress and maintain context continuity without redundant inquiries.
- **Schedule:** This category contains structured, time-sensitive data representing the user’s global timeline. It includes explicit appointments, deadlines, and recurring routines (e.g., "Gym every Tuesday at 6 PM"). The primary function of this memory type is to facilitate

Academic Writing
Topic ideation, literature synthesis, structural outlining, argumentation refinement, and stylistic editing.
Career Development
CV/Resume polishing, mock interview simulation, behavioral question strategy, and career narrative refinement.
Code Architecture Design
System modularization, design pattern application, API interface definition, technical stack selection, and scalability planning.
Financial Planning
Long-term goal formulation, asset allocation, risk assessment, and financial literacy education.
Fitness
Personalized workout routines, dietary tracking, biometric monitoring, and progressive plan adaptation.
Health Consultation
Symptom triage, medical report interpretation, lifestyle health advisory, and chronic condition management.
Knowledge Learning
Curriculum scaffolding, progress tracking, knowledge reinforcement, and adaptive learning path design.
Literary Creation
Narrative ideation, world-building coherence, character development, and stylistic enhancement.
Mental Health Support
Emotional support, cognitive reframing strategies, interpersonal conflict resolution, and stress management.
Project Management
Task decomposition, resource scheduling, documentation maintenance, and retrospective analysis.
Travel Planning
Multi-day itinerary scheduling, POI recommendations, culinary integration, and dynamic constraint-based adjustments.

Table 8: Overview of the 11 evaluation scenarios and their core task attributes in REALMEM.

temporal reasoning and conflict detection, ensuring that proposed project plans do not violate existing temporal constraints.

sonalization, empathy, and strict adherence to context-specific constraints.

A.1.3 Definition of Topic Types

To ensure the dataset reflects the diversity and complexity of real-world human-AI interactions, we curated eleven high-frequency user scenarios. As detailed in Table 8, these scenarios span multiple distinct domains:

- **Life Planning & Management:** Scenarios like *Travel Planning*, *Financial Planning*, and *Project Management* that require the agent to handle dynamic scheduling, resource allocation, and hierarchical goal decomposition.
- **Professional & Skill Development:** Tasks such as *Academic Writing*, *Literary Creation*, *Career Development*, *Knowledge Learning*, and *Code Architecture Design*, focusing on iterative content refinement, technical precision, and long-term learning trajectories.
- **Personal Well-being:** Sensitive domains including *Fitness*, *Mental Health Support*, and *Health Consultation*, which demand high per-

For each scenario, we explicitly modeled core attributes—ranging from budget constraints to emotional coping mechanisms—to establish clear evaluation boundaries.

A.2 Supplementary Experiments

This section presents additional experiments to validate the robustness and generalizability of RealMem across different backbone models and evaluation judges.

A.2.1 Open-Source Backbone Evaluation

To demonstrate that RealMem is not limited to proprietary models, we replace the GPT-4o backbone with DeepSeek-V3 for the A-mem system. As shown in Table 9, A-mem with DeepSeek-V3 achieves slightly higher scores than with GPT-4o, yet still falls substantially short of the Oracle upper bound. This indicates that simply upgrading the LLM backbone yields limited gains, and that optimizing the memory system architecture itself remains the critical bottleneck.

Method	QA Score	Mem Recall	Mem Helpful
<i>Backbone: DeepSeek-V3</i>			
A-mem (Session / Top-5)	0.6311	0.5918	0.6594
A-mem (Memory / Top-20)	0.5206	0.5091	0.5829
<i>Backbone: GPT-4o</i>			
A-mem (Session / Top-5)	0.606	0.590	0.655
A-mem (Memory / Top-20)	0.492	0.455	0.513
Oracle (Session)	0.696	0.767	0.845
Oracle (Memory)	0.804	0.993	0.922

Table 9: Performance of A-mem under different backbone models. DeepSeek-V3 yields marginal improvements over GPT-4o, but a substantial gap to Oracle remains.

A.2.2 Cross-Model Judge Evaluation

To verify that our evaluation is not biased by the choice of GPT-4o as the judge model, we re-evaluate all system outputs using DeepSeek-V3 as an alternative judge. Tables 10 and 11 show results at the Session (Top-5) and Memory (Top-20) levels, respectively. While absolute QA scores differ slightly between the two judges, the relative performance rankings remain perfectly consistent across both settings, confirming that our evaluation metrics are robust and generalize well across different evaluator models.

System	GPT-4o Judge		DeepSeek-V3 Judge	
	QA Score	Rank	QA Score	Rank
Graph Mem	1.533	1	1.363	1
Mem0	1.531	2	1.291	2
A-mem	1.479	3	1.269	3

Table 10: Cross-model judge comparison at Session level (Top-5). Rankings are identical across both judges.

System	GPT-4o Judge		DeepSeek-V3 Judge	
	QA Score	Rank	QA Score	Rank
MemoryOS	1.352	1	1.138	1
Graph Mem	1.275	2	1.072	2
Mem0	1.214	3	1.068	3
A-mem	1.186	4	1.055	4

Table 11: Cross-model judge comparison at Memory level (Top-20). Rankings are identical across both judges.

A.3 Human Evaluation Protocol

To validate the reliability of our automatic evaluation metrics, we conducted a human evaluation study with the following protocol.

Sample Size. We randomly sampled 30 dialogue instances from the test set. Each instance contains a user query paired with responses from four systems (MemoryOS, Graph Memory, A-mem, and Mem0), forming 30 four-way comparative evaluation tasks. Each annotator independently evaluated 15 samples.

Annotators. Two annotators with NLP backgrounds independently completed the evaluation. All system identities were strictly blinded to the annotators prior to annotation to minimize subjective bias.

Annotation Guidelines. For each sample, annotators assigned an overall ranking (1–4, where 1 = best) to the four system outputs based on the following four dimensions:

- **Memory Accuracy:** Whether the response correctly references key information from dialogue history without hallucination or omission.
- **Response Relevance:** Whether the response directly addresses the user’s current query.
- **Information Completeness:** Whether the response fully leverages historical context to provide a thorough answer.
- **Linguistic Fluency:** Whether the response is natural and clearly expressed.

Ties were not permitted. When outputs were difficult to distinguish, **Memory Accuracy** served as the primary tiebreaker.

Agreement Metric. We adopt Kendall’s W (Coefficient of Concordance) to measure inter-annotator agreement on the ranking task, which is the standard metric for multi-system ranking scenarios (range: $[0, 1]$, higher = better agreement). The resulting Kendall’s $W \geq 0.75$ indicates strong agreement between the two annotators.

A.4 Comparison of QA Score and Recall across different topics for four methods.

Table 12 reveals a significant domain sensitivity across all evaluated architectures, characterized by a sharp performance stratification between rigid and entity-centric tasks. Specifically, *Code Architecture Design* consistently yields the lowest scores across all methods (e.g., MemoryOS QA:

Topic	A-Mem		Mem0		MemoryOS		Graph Memory	
	QA	Recall	QA	Recall	QA	Recall	QA	Recall
Academic Writing	0.383	0.504	0.517	0.554	0.550	0.552	0.450	0.458
Career Development	0.485	0.517	0.515	0.502	0.553	0.593	0.492	0.467
Code Architecture Design	<u>0.322</u>	<u>0.291</u>	<u>0.380</u>	<u>0.446</u>	<u>0.363</u>	<u>0.381</u>	<u>0.363</u>	<u>0.345</u>
Financial Planning	0.453	0.532	0.453	0.540	0.522	0.563	0.497	0.577
Fitness	0.418	0.423	0.482	0.548	0.476	0.510	0.503	0.489
Health Consultation	0.556	0.470	0.681	0.569	0.694	0.655	0.681	0.651
Knowledge Learning	0.432	0.438	0.445	0.519	0.484	0.511	0.482	0.481
Literary Creation	0.437	0.516	0.498	0.581	0.517	0.633	0.502	0.532
Mental Health Support	0.465	0.441	0.511	0.526	0.606	0.584	0.569	0.552
Project Management	0.420	0.506	0.447	0.488	0.533	0.526	0.453	0.438
Travel Planning	0.368	0.462	0.385	0.534	0.442	0.513	0.424	0.483

Table 12: Performance variability across topics. For each method, the best-performing topic is highlighted in **bold**, and the worst-performing topic is underlined. Comparisons are made column-wise (per method).

0.363), indicating that current retrieval-based memory systems struggle with engineering tasks demanding strict logical consistency and precise dependency tracking, where fuzzy semantic matching is often insufficient. In contrast, *Health Consultation* emerges as the most robust scenario (e.g., MemoryOS QA: 0.694), likely because medical dialogues revolve around distinct named entities (e.g., symptoms, medications) that serve as strong semantic anchors for retrieval. This divergence suggests that while current long-term memory paradigms are effective for open-ended, consultative interactions, they remain insufficient for problem-solving scenarios that require high-precision state maintenance.

A.5 Case Studies

To provide a clearer illustration of our defined problem types, we present the following case studies. These examples serve to concretize the abstract definitions, demonstrating how the agent must leverage distinct memory operations—ranging from incremental reasoning to conflict resolution—to address specific challenges in long-term interactions.

As illustrated in the case study below, the user acknowledges prior context (attractions and food suggestions) and explicitly initiates a state transition toward itinerary planning. The system leverages memory to confirm these preferences and proposes a logical continuation. This exemplifies *Static Retrieval*, where the dialogue builds con-

structively upon established history to advance the specific task goal.

Static Retrieval

► **User:**

Hi! I've had a look at the updated attractions list. The foodie suggestions are great! I'm excited to start mapping out the actual day-by-day itinerary now. What do we need to do next?

► **Assistant:**

That's fantastic to hear! I'm glad you liked the foodie suggestions.

Since we have the overall 'Classic Loop' itinerary and now a great list of local food experiences, the next logical step is exactly what you said: mapping out the actual day-by-day itinerary, making sure to weave in those delicious foodie stops.

Would you like to start by reviewing Day 1 and Day 2, or perhaps focus on specific regions first, like Queenstown or the West Coast, to integrate the food elements?

► **Memory Used:**

[Food Scheme] → [Application]: Acknowledge user's positive feedback on foodie suggestions and propose integrating them into the day-by-day itinerary.

As illustrated in the case study below, the user's input is predominantly emotional and affirmative, lacking explicit directives or queries. To maintain dialogue momentum, the assistant proactively retrieves a stored long-term priority (booking flights and RV rental) to propose the next logical step. This exemplifies *Proactive Alignment*, where the system utilizes memory to bridge the conversational gap and guide the planning flow when im-

mediate user intent is ambiguous.

Proactive Alignment

► **User:**

Wow, this is fantastic! You've really outdone yourself. The foodie additions are exactly what I was hoping for, and you've hit so many great spots. Fergburger is definitely going straight to the top of my list, and local salmon in Tekapo sounds absolutely amazing. I'm already imagining all the delicious food!

This enhanced guide sounds absolutely perfect, and yes, my taste buds are definitely ready to be tantalized!

► **Assistant:**

That's wonderful to hear! I'm so glad the foodie additions hit the mark and you're excited about all the delicious experiences awaiting you in New Zealand.

Now that we have a fantastic baseline for attractions and a mouth-watering foodie guide, we can move on to the next crucial step. You previously mentioned that booking flights and the RV rental would be the most important next priority to lock in. Shall we dive into those details now?

► **Memory Used:**

[Priority: Flights & RV] → [Application]: Propose the next step in trip planning, which is booking flights and RV rental, as confirmed by the user previously.

As illustrated in the case study below, the user introduces a constraint conflict: maintaining a fixed 12-day duration while integrating substantial new activities (West Coast). This necessitates a structural modification of the existing itinerary rather than a simple addition. The system retrieves specific plan details (e.g., Queenstown commitments) to evaluate the impact of resource reallocation. This exemplifies *Dynamic Updating*, where the agent leverages memory to negotiate trade-offs and alter the plan's state to accommodate conflicting requirements.

Dynamic Updating

► **User:**

That sounds amazing, I really want to do it! Hmm, extending the trip might be tough with my work schedule, so I'd really prefer to stick to 12 days if possible.

But I also don't want the whole trip to feel super rushed and stressful, you know? Can you show me what it would look like if we tried to fit the two West Coast days into the existing 12 days? I want to see how much more 'rushed' it would actually be before I decide.

► **Assistant:**

I understand completely! Sticking to 12 days while adding two full West Coast days is a challenge, but we can definitely map it out so you can see the trade-offs.

To incorporate two *full* days on the West Coast for your heli-hike and exploration, we'd need to allocate three nights in the Franz Josef/Fox Glacier area instead of the original one night. This means we need to free up two

days from elsewhere in the itinerary.

The most practical way to do this without drastically altering the core...

► **Memory Used:**

[Plan Detail: Queenstown Day 4] →

[Application]: Identify the existing commitments in Queenstown that would be impacted by reducing the time allocated there.

As illustrated in the case study below, the user proposes a recurring workout schedule for specific days. To validate feasibility, the system retrieves existing temporal constraints (Study Sessions) from memory. By cross-referencing the proposed days with occupied time slots to ensure no overlap, the agent performs constraint satisfaction. This exemplifies *Temporal Reasoning*, where the system processes chronological relationships to maintain schedule consistency.

Temporal Reasoning

► **User:**

Hello, I want to arrange a workout schedule for Tuesdays, Thursdays, and Saturdays. How about that?

► **Assistant:**

Yes, that sounds excellent! I've checked your proposed times against your study sessions, and they fit perfectly without any overlaps.

So, to confirm, your fitness schedule will be:

- **Tuesdays:** 6:00 PM - 6:45 PM
- **Thursdays:** 6:00 PM - 6:45 PM
- **Saturdays:** 9:00 AM - 9:45 AM

Now that we have the times locked in, the next step is to outline the specific exercises...

► **Memory Used:**

[Constraints: Study Sessions (Tue/Thu 20:00-21:30, Sat 10:00-12:00)] →

[Validation]: Check for potential time conflicts with newly proposed workout days.

A.6 Prompts

This section presents some of the important prompt templates involved in the paper.

A.6.1 Evaluation Prompts

Figures 5 and 6 illustrate the specific prompt templates used in our evaluation framework for assessing response consistency and memory retrieval quality, respectively.

Prompt for Response Consistency Evaluation

Instruction: Your task is to evaluate the consistency between the [candidate answer] and the [user-related memory].

Input Data:

- User's current query
- User-related memory (representing the latest valid user state)
- Reference answer (based on the relevant memory)
- Candidate answer (to be evaluated)

Evaluation Rules:

- Focus only on whether “facts, constraints, preferences, and confirmed states” are correctly used.
- Do NOT evaluate language style, tone, politeness, empathy, or fluency.
- Do NOT give a high score just because the answer “sounds reasonable”.
- The reference answer is only to help understand how relevant memory should ideally be used; a candidate answer does not need to exactly match the reference answer to receive a full score.

Scoring Criteria:

- **Score 0 (Poor):** The candidate answer conflicts with the user-related memory.
- **Score 1 (Fair):** The candidate answer does not conflict with the relevant memory but is generic and not based on user memory.
- **Score 2 (Good):** The candidate answer uses part of the user-related memory.
- **Score 3 (Very good):** The candidate answer (like the reference answer) uses all of the user-related memory.

Output Format:

```
{
  "score": int,
  "reason": str # Briefly explain the reason for the score
}
```

Figure 5: The specific prompt used for evaluating the consistency between the generated response and user memory.

Prompt for Memory Retrieval Quality Evaluation

Instruction: Your task is to evaluate the consistency between the [retrieved memory] and the [ground-truth memory], and whether the retrieved memory is helpful.

Input Data:

- <question>: The user's current query.
- <groundtruth_memory>: The true memory that is helpful for answering the question.
- <retrieved_memory>: The retrieved memory.

Evaluation Dimensions:

1. **Memory Recall** (0–1): Semantics-aware memory recall calculation.
 - *step1*: For each groundtruth_memory, check in sequence whether its semantics are contained in any retrieved_memory.
 - *step2*: Count how many groundtruth_memory items are covered (hits_cnt).
 - *step3*: Compute the final recall score as hits_cnt / total number of groundtruth_memory items.
2. **Memory Helpfulness** (0–2): The helpfulness of the retrieved memory.
 - **Score 0**: retrieved_memory contains mutually conflicting or contradictory memories, which not only fail to help answer the question but may also cause confusion.
 - **Score 1**: retrieved_memory is somewhat helpful for answering the question (can provide partial supporting evidence).
 - **Score 2**: retrieved_memory is very helpful for answering the question (can provide comprehensive supporting evidence).

Output Format: Please provide your evaluation results using the following structure:

```
{
  "Mem_recall": float,
  "Mem_helpful_score": int,
  "Mem_hits": list[str], # List the matched groundtruth_memory items
  "Mem_helpful_reason": str # Explain the reason for the score
}
```

Figure 6: The specific prompt used for evaluating the quality of memory retrieval, focusing on semantic recall and helpfulness metrics.