Beyond Dialogue Time: Temporal Semantic Memory for Personalized LLM Agents

Miao Su[1,2,3,4], Yucan Guo[1,2,3], Zhongni Hou[4], Long Bai[1,2], Zixuan Li[1,2], Yufei Zhang[4], Guojun Yin[4], Wei Lin[4], Xiaolong Jin[1,2,3]*, Jiafeng Guo[1,2,3], Xueqi Cheng[1,2,3],

[1]Institute of Computing Technology, Chinese Academy of Sciences
[2]State Key Laboratory of AI Safety
[3]School of Computer Science, University of Chinese Academy of Sciences
[4]Meituan

sumiao22z@ict.ac.cn

Abstract

Memory enables Large Language Model (LLM) agents to perceive, store, and use information from past dialogues, which is essential for personalization. However, existing methods fail to properly model the temporal dimension of memory in two aspects: 1) Temporal inaccuracy: memories are organized by dialogue time rather than their actual occurrence time; 2) Temporal fragmentation: existing methods focus on point-wise memory, losing durative information that captures persistent states and evolving patterns. To address these limitations, we propose Temporal Semantic Memory (TSM), a memory framework that models semantic time for point-wise memory and supports the construction and utilization of durative memory. During memory construction, it first builds a semantic timeline rather than a dialogue one. Then, it consolidates temporally continuous and semantically related information into a durative memory. During memory utilization, it incorporates the query's temporal intent on the semantic timeline, enabling the retrieval of temporally appropriate durative memories and providing time-valid, duration-consistent context to support response generation. Experiments on LongMemEval and LoCoMo show that TSM consistently outperforms existing methods and achieves up to 12.2% absolute improvement in accuracy, demonstrating the effectiveness of the proposed method.

1 Introduction

Recent years have witnessed the rapid emergence of Large Language Model (LLM) agents, autonomous systems built upon LLMs with capabilities for reasoning, tool use, and long-term interaction (Matarazzo and Torlone, 2025; Minaee et al., 2025; Luo et al., 2025). A key component of LLM agents is *memory*. Instead of changing model parameters, memory provides an explicit store of interaction-derived information that agents can later reuse, enabling adaptation over extended interactions (Sumers et al., 2023; Wu et al., 2025b). Personalized dialogue agents (Chhikara et al., 2025a; Li et al., 2025) exemplify this setting. Memory helps the agent maintain user-specific context, such as past plans, preferences, and evolving situations. It is typically implemented as a three-stage pipeline of construction, update, and utilization (Hu et al., 2025), so that responses remain coherent, grounded in prior interactions, and consistent over time.

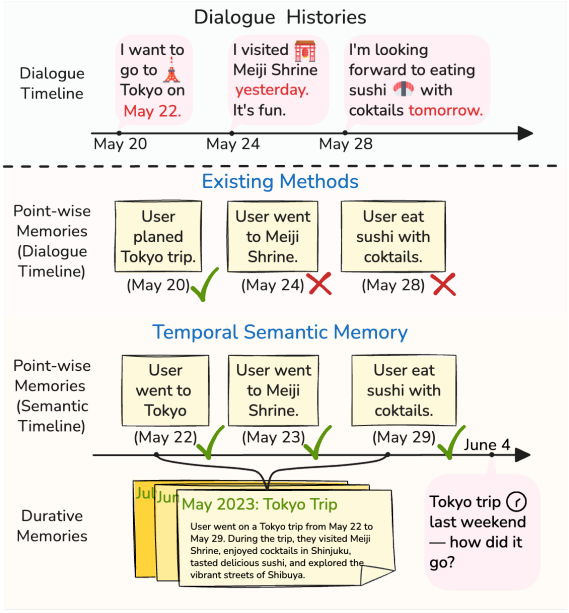

Figure 1: Comparison of existing methods vs. TSM with semantic timeline and durative memories.

Recent works construct memory by organizing dialogue histories into structured memory entries and retrieving them when needed (Zhong et al., 2023; Kim et al., 2025; Tan et al., 2025; Sun and Zeng, 2025). However, as shown in Figure 1, existing methods suffer from two critical limitations in how they model temporal information. (1) **Temporal inaccuracy.** Most systems treat the *dialogue*

* Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 29935–29951
July 2-7, 2026

timeline (when a chat turn is produced) as the primary temporal signal when assessing recency or relevance (Rasmussen et al., 2025a). This is problematic because users often talk about events that occur at different times than the conversation itself, including future plans, past trips, and ongoing states. For example, when a user talks on May 28 about a trip happening on May 29, dialogue time and event time are misaligned; using the dialogue timeline alone can cause the system to store or retrieve memories under the wrong time context. (2) **Temporal fragmentation.** Many methods store memories as isolated, point-wise entries (Tan et al., 2025; Fang et al., 2025; Chhikara et al., 2025b; Rasmussen et al., 2025a). This representation breaks temporally continuous experiences into disconnected records, making it difficult to recover durations and long-term states. For instance, the multiple entries within a week in Figure 1 together describe a coherent Tokyo trip; treating them independently ignores their temporal continuity and semantic relatedness, which in turn hinders the formation of persistent states and evolving patterns. Together, these issues prevent agents from retrieving complete and relevant memories, especially when the user query implicitly assumes a coherent real-world timeline.

In contrast, human memory relies on time as a scaffold for ordering and linking real-life experiences, supporting coherent recall across long-term memories (MacDonald et al., 2011; Huet et al., 2025). This highlights the importance of modeling time beyond a point-wise, dialogue-timeline view.

To this end, we propose **Temporal Semantic Memory (TSM)**, a memory framework designed to support semantic-time grounded and duration-aware memory access. In particular, TSM addresses the above challenges with two tightly coupled components: (1) **Duration-aware memory construction.** TSM builds a *semantic timeline* through a temporal knowledge graph, aligning memory with when events happen and how long they last. Beyond recording point-wise facts, TSM links temporally continuous and semantically related mentions and consolidates them into *durative summaries* that capture long-term states (i.e., topics and personas). (2) **Semantic-time guided memory utilization.** During memory utilization, TSM incorporates the query's semantic temporal intent and retrieves memories at the appropriate temporal granularity, rather than relying on dialogue-time recency or semantics-only similarity that disregards timing. This enables the system to return time-valid, duration-consistent context for response generation with correct temporal grounding. In addition, TSM maintains memory with a lightweight hierarchical mechanism: it incrementally updates temporal facts online and periodically consolidates summaries to improve long-term consistency.

Extensive experiments on LONGMEMEVAL (Wu et al., 2025a) and LOCOMO (Maharana et al., 2024) demonstrate that TSM consistently outperforms strong memory baselines, with the largest gains on multi-session understanding and temporal reasoning tasks, validating the effectiveness of semantic-time grounding and duration-aware consolidation.

2 Related Work

2.1 Agent Memory

LLM agents are increasingly equipped with long-term memory that grows and adapts over time, allowing them to accumulate knowledge, recall prior context, and adjust behavior based on experience (Camel-AI, 2025; Liang et al., 2025; Google, 2025; ByteDance, 2025). There are several functions of agent memory: Factual memory stores persistent information such as user profiles, dialogue history, and world facts to support long-term consistency and personalization (Wu et al., 2026), as explored in memory-augmented dialogue agents and long-term user modeling systems (Park et al., 2023; Packer et al., 2024; Nan et al., 2025; Kwon et al., 2025); experiential memory records past interaction trajectories and distilled strategies to enable continual self-improvement across tasks, exemplified by case-based, strategy-based, and skill-based learning in reflective and self-improving agents (Zhang et al., 2026; Shinn et al., 2023; Yan et al., 2025; Zhou et al., 2025; Ouyang et al., 2025); working memory provides mechanisms for the active management of transient context (Zhou et al., 2025; Zhang et al., 2025b).

In this work, we focus on user-specific factual memory to support temporally grounded, personalized agent over long-term interactions.

2.2 Graph-structured Memory

In the context of agent memory, graph-structured memory arises naturally when agents accumulate relational insights over time (Yang et al., 2026). Mem0^g employs a scalable two-phase architecture (extraction and update) that dynamically

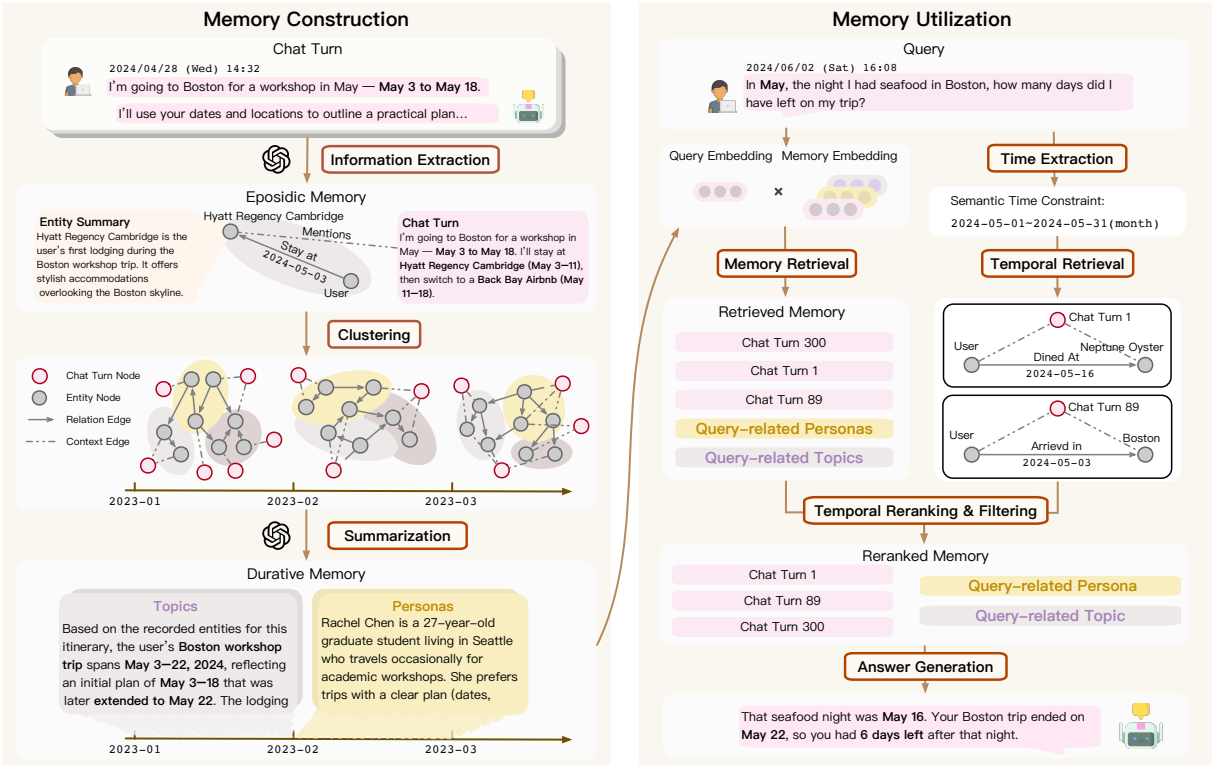

Figure 2: The overall framework of TSM. Memory consolidation constructs a temporal knowledge graph from episodic memory and subsequently consolidates it into time-aware durative memory. Memory utilization retrieves accurate memories by applying semantic-temporal constraints.

stores and retrieves salient facts in a knowledge graph (Chhikara et al., 2025a). A-MEM (Xu et al., 2025) builds an interconnected memory network inspired by Zettelkasten (Kadavy, 2021), where each new memory is represented as a structured note with attributes (e.g., keywords/tags), and the system dynamically links related memories and updates existing notes as new information arrives, enabling an evolving memory graph. Zep (Rasmussen et al., 2025a) proposes a temporal knowledge graph memory layer, emphasizing temporal reasoning over an evolving graph rather than static document retrieval. However, it ignores time during the memory retrieval stage. Other recent systems (Zhang et al., 2025a; Wu et al.) similarly allow the graph to be constructed, extended, or reorganized throughout the agent's operation.

Many prior graph-based memory systems primarily treat the graph as a persistent store of extracted facts, but the *evolution* of memory in a time period is not considered, yet the retrieval stage still underutilizes the *semantic time* encoded in the graph, resulting in temporally misaligned recall.

3 Methodology

3.1 Preliminary

We consider the task of building a personalized dialogue agent in a **multi-session** conversational setting. A *session* represents a distinct interaction period, often delimited by user inactivity, explicit user confirmation of conversation completion, or the initiation of a new dialogue thread. Within each session, the conversation unfolds as a sequence of chat turns, where a *chat turn* consists of a user query and the agent's corresponding response.

Typically, a standard agent memory system implements three core functions:

(1) Construction, which transforms raw chat history D into structured memory M,

$$M = f_{\text{construct}}(D)$$

(2) Update, which refines the existing memory with new conversational data D',

$$M' = f_{\text{update}}(M, D')$$

(3) Utilization (Retrieval), which generates an answer A to a given query Q based on the stored memory,

$$A = f_{\text{retrieve}}(M, Q).$$

3.2 Overview

As illustrated in Figure 2, TSM consists of two stages: Memory Construction (§3.3) that builds a temporal knowledge graph as a episodic memory and organizes episodic interactions into durative memories; Memory Utilization (§3.4) that performs constraint-aware retrieval by integrating dense matching with temporal reranking and filtering for accurate memory access. Additionally, we build an Update Mechanism (§3.5) that periodically refreshes both the episodic memory and the constructed durative memory to ensure long-term consistency.

3.3 Duration-aware Memory Construction

We construct two complementary memory types from dialogue streams through a hierarchical process:

Episodic Memory captures atomic facts with specific semantic timestamp (e.g., "visited Tokyo on May 23"), organized as a temporal knowledge graph with explicit temporal grounding.

Durative Memory maintains enduring patterns that persist across extended periods, derived from episodic memory through temporal segmentation and semantic abstraction. Unlike event-specific episodic memories, durative memories represent stable user characteristics (e.g., sustained interests, evolving preferences) extracted from accumulated experiences. This concept parallels semantic memory in cognitive science.

3.3.1 Episodic Memory Construction

We construct a Temporal Knowledge Graph (TKG) as a structured memory index that records episodic information mentioned in the dialogue history following Zep (Rasmussen et al., 2025b). The TKG is not directly used as retrievable memory content; instead, it provides precise temporal localization and semantic time access for subsequent memory construction and retrieval.

Formally, the TKG is defined as a set of temporally grounded facts

$$\mathcal{G} = \{(e_s, r, e_o, t) \mid t \in \mathbb{T}\}. \tag{1}$$

where e_s and e_o denote the subject and object entities, r denotes a semantic relation, and t denotes the time point when the fact is valid.

In addition to temporal facts, each entity node maintains a compact entity summary extracted from its supporting dialogue contexts. We represent an entity as $e \triangleq (n_e, s_e)$, where n_e is the canonical entity name and s_e is an LLM-generated entity summary that consolidates salient attributes of e.

We extract entities and relations from each turn (with a context window of the preceding n turns) and incrementally integrate them into the TKG via deduplication and temporal consistency checks (details in Section 3.5).

The resulting TKG serves as an episodic memory index that supports time-aware access, temporal filtering, and consistency checking, while deferring semantic abstraction to later stages.

3.3.2 Durative Memory Construction

Based on the episodic memory graph, we construct durative memory by aggregating episodic information into higher-level semantic representations.

Given the temporal knowledge graph, we partition it into a sequence of temporal slices

$$\begin{aligned} \mathcal{G} &= \bigcup_k \mathcal{G}^{(k)}, \\ \mathcal{G}^{(k)} &= \{(e_s, r, e_o, t) \mid t \in [\tau_k, \tau_{k+1})\}, \end{aligned} \tag{2}$$

where $[\tau_k, \tau_{k+1})$ denotes a fixed temporal interval. In our implementation, the granularity is set to one month by default.

For each temporal slice $\mathcal{G}^{(k)}$, we collect the involved entity set

$$\mathcal{E}^{(k)} = \{e \mid e \text{ appears in } \mathcal{G}^{(k)}\}. \tag{3}$$

We apply a Gaussian Mixture Model (GMM) (Huang et al., 2025) to cluster entities within the same temporal slice

$$p(z \mid e) = \mathrm{GMM}(\mathbf{h}_e^{\mathrm{name}}), \tag{4}$$

where z denotes a latent cluster capturing a coherent semantic theme. For each entity e, we assign it to the most likely cluster $a(e) = \arg\max_z p(z \mid e)$, and define

$$\mathcal{E}_z = \{e \in \mathcal{E}^{(k)} \mid a(e) = z\}. \tag{5}$$

For each cluster z in the k-th temporal slice, let

$$\mathcal{X}_z = \{(n_e, s_e) \mid e \in \mathcal{E}_z\} \tag{6}$$

denote the entity-summaries in the cluster. We define a topic as

$$\begin{aligned} \mathrm{Topic}_z &= \{\tau_k, s_z, \mathbf{c}_z\}, \\ s_z &= \mathrm{LLM}_{\mathrm{sum}}(\mathcal{X}_z), \\ \mathbf{c}_z &= \mathrm{Embedding}(s_z), \end{aligned} \tag{7}$$

where τ_k denotes the corresponding time slice, s_z is the textual topic summary, and $\mathbf{c}_z$ is its embedding used for downstream retrieval.

To capture user-level characteristics, we further aggregate the dialogue contexts associated with the entities in $\mathcal{E}_z$. Using a bidirectional index between entities and their originating chat turns, we collect the corresponding dialogue

$$\mathcal{D}_z = \{d \mid d \text{ mentions } e,\ e \in \mathcal{E}_z\}. \tag{8}$$

The persona representation is defined as

$$\begin{aligned} \text{Persona}_z &= \{\tau_k, \mathbf{p}_z, \mathbf{u}_z\}, \\ \mathbf{p}_z &= \text{LLM}_{\text{sum}}(\mathcal{D}_z), \\ \mathbf{u}_z &= \text{Embedding}(\mathbf{p}_z), \end{aligned} \tag{9}$$

where $\mathbf{p}_z$ captures stable user traits, preferences, and behavioral patterns expressed within the temporal slice, and $\mathbf{u}_z$ denotes its embedding.

Through temporal segmentation and semantic abstraction, the constructed topics and personas form hierarchical, temporally anchored durative memory. It captures durative user states beyond isolated point events, thereby supporting efficient long-term storage and subsequent constraint-aware retrieval.

3.4 Semantic-time Guided Memory Utilization

This stage retrieves memory that is both semantically relevant and temporally consistent with the user query. Given a query q, we (i) infer its semantic time constraint T_q, (ii) perform dense retrieval over topics, personas, and raw dialogue chunks, and (iii) enforce the temporal constraint by filtering time-anchored summaries and promoting candidates supported by temporally valid evidence from the TKG.

Given a user query q issued at time t_{now}, we first parse its semantic-time constraint T_q, i.e., the time range when the described event is intended to hold (rather than the dialogue time)

$$T_q = \text{ParseTime}(q, t_{\text{now}}), \tag{10}$$

where $\text{ParseTime}(\cdot)$ is implemented with spaCy (Honnibal et al., 2020) and resolves both explicit and relative time expressions.

Let the retrievable memory pool be

$$\mathcal{M} = \mathcal{M}_{\text{topic}} \cup \mathcal{M}_{\text{persona}} \cup \mathcal{M}_{\text{raw}}, \tag{11}$$

where each topic/persona entry m carries a slice timestamp $\tau(m)$ from construction, while each raw segment $m \in \mathcal{M}_{\text{raw}}$ is a chat turn. We compute dense retrieval scores

$$s_{\text{sem}}(m; q) = \text{sim}\big(\text{Enc}(q), \text{Enc}(m)\big), \tag{12}$$

and retrieve Top-K candidates

$$\mathcal{D} = \text{TopK}_{m \in \mathcal{M}}\ s_{\text{sem}}(m; q). \tag{13}$$

To align retrieval with the temporal intent in q, we apply temporal filtering to the retrieved topics/personas (post-retrieval in our implementation):

$$\text{Keep}(m, T_q) = \begin{cases} \mathbb{I}\big[\tau(m) \in T_q\big], & m \in \mathcal{M}_{\text{topic}} \\ & \cup\, \mathcal{M}_{\text{persona}}, \\ 1, & m \in \mathcal{M}_{\text{raw}}. \end{cases} \tag{14}$$

In parallel, we query the TKG for temporally valid facts and map them to their originating chat turns via the bidirectional index:

$$\begin{aligned} \mathcal{F}_T &= \{(e_s, r, e_o, t) \in \mathcal{G} \mid t \in T_q\}, \\ \mathcal{S}_T &= \text{Idx}(\mathcal{F}_T). \end{aligned} \tag{15}$$

where $\text{Idx}(\cdot)$ returns the set of raw chat turns linked to the facts.

Finally, we rerank candidates through a composite scoring function that prioritizes semantic-time alignment before semantic similarity. To be specific, we use the indicator $\mathbb{I}[\tau(m) \in T_q]$ as the primary key and $s_{\text{sem}}(m; q)$ as the secondary key:

$$\pi(m; q) = \Big(\mathbb{I}\big[\tau(m) \in T_q\big],\ s_{\text{sem}}(m; q)\Big), \tag{16}$$

and sort candidates in descending lexicographic order of $\pi(\cdot)$:

$$\mathcal{R} = \text{Sort}_{m \in \mathcal{D}}\ \pi(m; q). \tag{17}$$

This design enforces the query time constraint on compact summaries (topics/personas) while using TKG-grounded evidence that anchored on the Semantic timeline to promote relevant chat turns, yielding more accurate and contextually grounded retrieval.

3.5 Hierarchical Memory Update

Memory in TSM is maintained via a dual-stage update mechanism that separates low-latency graph maintenance from high-cost duration consolidation.

Table 1: Category-wise Accuracy For LONGMEMEVAL_S. Accuracy (%) by method across question types. Parentheses indicate category proportion and sample size. For the frontier-model variants, memory construction remains unchanged and uses GPT-4o-mini for graph extraction; only the answer generation model is replaced with the corresponding frontier model.

Method	**ACC (%)**	**Temporal** (n=133)	**Multi-Session** (n=133)	**Knowledge-Update** (n=78)	**Single-User** (n=70)	**Single-Assistant** (n=56)	**Single-Preference** (n=30)
GPT-4o-mini							
Full Text	56.80	31.58	45.45	76.92	87.14	89.29	36.67
Naive RAG	61.00	39.85	48.48	67.95	90.00	**98.21**	53.33
LangMem	37.20	15.79	20.30	66.67	60.00	46.43	60.00
A-MEM	62.60	47.36	48.87	64.11	**92.86**	96.43	46.67
Zep	60.20	36.50	47.40	76.90	81.40	81.80	30.00
MemoryOS	44.80	32.33	31.06	48.72	80.00	64.29	30.00
Mem0	53.61	40.15	46.21	70.12	81.43	41.07	**60.00**
TSM	**74.80**	**69.92**	**69.17**	**80.77**	87.14	94.64	40.00
Qwen3-30B-A3B-Instruct-2507							
Full Text	54.80	33.08	35.61	76.92	82.86	87.50	50.00
Naive RAG	60.80	36.84	47.73	65.38	91.43	**98.21**	70.00
LangMem	50.80	37.60	38.35	67.95	78.57	42.86	70.00
A-MEM	65.20	51.88	51.12	76.93	90.00	96.43	40.00
MemoryOS	49.60	28.57	36.84	61.54	72.86	92.86	33.33
Mem0	39.51	41.94	28.13	28.57	55.32	26.09	**81.82**
TSM	**74.80**	**63.91**	**63.91**	**82.05**	**97.14**	92.86	66.67
TSM with Frontier Models							
TSM + *Gemini-3-Pro*	74.60	60.90	69.17	**93.59**	92.86	**91.07**	36.67
TSM + *Gemini-3-Flash*	**79.60**	**71.43**	**75.19**	92.31	92.86	89.29	**53.33**

Lightweight Online Graph Update. As new dialogue turns arrive, we update the TKG as the *episodic memory* in an incremental manner, without blocking online inference. For each extracted candidate entity, *add* creates a new node when the mention corresponds to an unseen entity; *merge* maps the mention to an existing node and integrates newly observed attributes or contextual evidence to enrich the entity representation while avoiding duplication. Relations are updated in a temporally grounded way that each fact is associated with a *valid_time* and an *invalid_time*, indicating the interval during which it holds. Given a new extracted fact, we compare it with existing edges in both semantics and time, and apply one of four operations: *DUPLICATE*, *ADD*, *INVALIDATE*, and *UPDATE*. These lightweight operations keep the index chronologically faithful and consistent as interactions evolve.

Sleep-time Summary Consolidation. In contrast to the memory graph, topic/persona summaries are high-level durative memories and are expensive to refresh. We therefore update summaries periodically (e.g., once per month, aligned with the summary time granularity) or when the accumulated turns exceed a preset threshold. During consolidation, we reorganize all entity mentions via GMM-based clustering and re-summarize the resulting clusters into updated topic and persona snapshots. This "sleep-time" procedure reduces construction cost and latency while preserving coherence over long horizons.

4 Experiments

In this section, we evaluate TSM on real-world datasets to assess its performance.

4.1 Experimental Setup

Dataset. The performance of TSM was evaluated on two public long-term memory benchmarks: LONGMEMEVAL and LOCOMO. LONGMEMEVAL (Wu et al., 2025a) is a comprehensive benchmark for assessing the long-term memory capabilities of chat assistants. It consists of 500 manually created questions to test five core memory abilities: information extraction, multi-session reasoning, temporal reasoning, knowledge updates, and abstention. Each question requires recalling information hidden within one or more task-oriented dialogues between a user and an assistant. We

Table 2: Category-wise accuracy on LoCoMo. Accuracy (%) of different memory systems across question types. For frontier-model variants, the memory graph is still constructed by GPT-4o-mini, while the final response is generated by the corresponding frontier model. Parentheses indicate category proportion and sample size.

Method	ACC (%)	Temporal (n=321)	Multi-Hop (n=282)	Open-Domain (n=96)	Single-Hop (n=841)
GPT-4o-mini					
Full Text	71.83	**76.92**	87.14	89.29	36.67
Naive RAG	63.64	67.95	90.00	**98.21**	53.33
LangMem	57.20	66.67	60.00	46.43	60.00
A-MEM	64.16	64.11	**92.86**	96.43	46.67
MemoryOS	58.25	48.72	80.00	64.29	30.00
Mem0^g	68.44	58.13	47.19	75.71	65.71
Zep	58.44	61.65	71.79	88.57	**91.07**
TSM	**76.69**	71.03	66.67	58.33	84.30
Qwen3-30B-A3B-Instruct-2507					
Full Text	**74.87**	**76.92**	82.86	87.50	50.00
Naive RAG	66.95	65.38	**91.43**	**98.21**	70.00
LangMem	60.53	67.95	78.57	42.86	70.00
A-MEM	56.10	76.93	90.00	96.43	40.00
MemoryOS	61.04	61.54	72.86	92.86	33.33
Mem0	43.31	28.57	55.32	26.09	**81.82**
TSM	71.23	65.42	64.54	56.25	77.41
TSM with Frontier Models					
TSM + *Gemini-3-Pro*	66.30	63.86	50.71	46.88	74.67
TSM + *Gemini-3-Flash*	73.96	**77.26**	57.45	51.04	80.86

utilize LongMemEvalS, a version where each question has approximately 115k tokens as its history. LoCoMo (Maharana et al., 2024) focuses on extremely long multi-session dialogues, containing 1,986 questions in five distinct categories: single-hop, multi-hop, temporal, open-domain, and adversarial reasoning.

Evaluation Metrics. We report **Accuracy (ACC)** for effectiveness, defined as the proportion of correctly answered questions. Following prior work, evaluation is conducted with *GPT-4.1-mini* as an LLM judge, guided by a detailed evaluation prompt (see Appendix A.4).

Baselines. We compare TSM against several representative baselines of conversational memory modeling. (1) Full Text, (2) Naive RAG, (3) LangMem (LangChain), (4) A-MEM (Xu et al., 2025), (5) MemoryOS (Kang et al.), (6) Mem0 or Mem0^g (a graph variant of Mem0) (Chhikara et al., 2025a), (7) Zep (Rasmussen et al., 2025a). In addition, all methods use GPT-4o-mini (OpenAI, 2024) and Qwen3-30B-A3B-Instruct-2507 (Qwen Team, 2025) as the LLM backbones.

4.2 Main Results

Tables 1 and 2 report the category-wise performance on LongMemEval_S and LoCoMo, respectively. Across both datasets, our method consistently achieves strong performance, outpacing existing baselines and demonstrating robust memory reasoning capabilities in diverse long-context scenarios.

LongMemEval. On LongMemEval_S, TSM achieves the highest overall accuracy of 74.80%, surpassing the A-MEM baseline (62.60%) on GPT-4o-mini. We set new state-of-the-art results on *Temporal*, *Multi-Session*, and *Knowledge-Update* questions, all of which are strongly dependent on time. The notable improvement in *Multi-Session* accuracy (+20.30%) underscores the crucial role of durative memory. By maintaining long-term contextual information, TSM enables more coherent and accurate reasoning across multiple interactions. The improvement in *Temporal* accuracy (+22.56%) highlights the effectiveness of using a semantic timeline. By leveraging temporal semantics, TSM can retrieve and update context across time, leading to more accurate handling of time-sensitive queries. Performance on *Single-Session*

Table 3: Ablation results on LONGMEMEVAL_S. Best numbers in each column are in bold. Δ denotes the absolute change in accuracy points, and Rel.% denotes the relative change with respect to TSM.

Name	Overall	Single-Session User	Temporal	Knowledge Update	Multi-Session	Single-Session Preference	Single-Session Assistant
TSM	**74.80**	87.14	**69.92**	80.77	69.17	**40.00**	94.64
w/o Temporal	72.80	87.14	63.91	79.49	**71.43**	33.33	91.07
Δ	-2.0	$+0.0$	-6.0	-1.3	$+2.3$	-6.7	-3.6
Rel.%	$\downarrow$ 2.7%	0.0%	$\downarrow$ 8.6%	$\downarrow$ 1.6%	$\uparrow$ 3.3%	$\downarrow$ 16.7%	$\downarrow$ 3.8%
w/o Persona/Summary	73.40	**88.57**	65.41	**82.05**	69.92	23.33	**96.43**
Δ	-1.4	$+1.4$	-4.5	$+1.3$	$+0.8$	-16.7	$+1.8$
Rel.%	$\downarrow$ 1.9%	$\uparrow$ 1.6%	$\downarrow$ 6.5%	$\uparrow$ 1.6%	$\uparrow$ 1.1%	$\downarrow$ 41.7%	$\uparrow$ 1.9%

Preference questions is lower; however, the limited sample size and high variance suggest that this category does not significantly impact overall performance. TSM also achieves superior performance on Qwen3 backbones, with an accuracy of 74.80%, and yields the best results on *Single-User* questions, showing strong performance across all categories.

LOCOMO. On LOCOMO, the full-text baseline achieves the highest performance on Qwen3-30B-A3B-Instruct-2507. This is expected because LOCOMO conversations are relatively short (16k to 26k tokens) compared to LONGMEMEVAL_S (115k tokens), making it feasible to fit the entire context within the model's window. More importantly, LOCOMO does not effectively test critical memory capabilities such as knowledge updates. When the full conversation is short enough to process directly, providing complete context naturally outperforms any retrieval-based approach that risks information loss.

Nevertheless, among all memory-based methods, TSM achieves the best performance with 71.23% accuracy on Qwen3-30B and 76.69% on GPT-4o-mini, substantially outperforming Naive RAG (63.64%) and Mem0^g (68.44%). Our method excels particularly on *Single-Hop* and *Temporal* questions, demonstrating that our temporal grounding and hierarchical memory organization provide superior retrieval precision compared to existing memory systems.

4.3 Efficiency

Table R2 reports both token cost and recall latency on LONGMEMEVAL_S. Although TSM does not use the fewest tokens overall, it achieves the best task performance (74.80% ACC). It also delivers the fastest recall latency, with a P50/P95 of 1.57/2.39 seconds.

These results suggest that the efficiency advantage of TSM mainly comes from its online retrieval design, rather than minimizing total token usage. Specifically, expensive memory construction is shifted to the update stage. During inference, the model only performs temporal filtering over a pre-built memory graph and local reranking over retrieved candidates. This avoids additional LLM-based extraction during recall. Moreover, the sleep-time update variant further reduces the total token cost from 2065.54k to 1959.53k. This shows that periodic offline summarization can reduce update overhead without affecting online latency. Overall, the results indicate that TSM offers a strong trade-off between effectiveness and practical inference efficiency for long-horizon memory retrieval.

4.4 Ablation Study

To evaluate the effectiveness of TSM, we perform an ablation study on LONGMEMEVAL_S under two settings. For the semantic timeline, the "w.o. temporal" variant removes temporal ranking and filtering during memory utilization and relies solely on dense retrieval. As a result, durative summaries cannot be selected based on the query's temporal intent, and semantically and temporally related chat turns are not prioritized. For durative memories, "w.o. summary" removes all topics and personas while retaining temporal reranking, ensuring that this configuration is not RAG-based.

As shown in Table 3, removing temporal retrieval ("w.o.temporal") leads to a noticeable degradation in overall performance (74.8% $\rightarrow$ 72.8%, -2.0), with the largest drop observed in *Temporal* questions (-6.0). This highlights the importance of explicit temporal modeling for addressing time-related queries. It also negatively impacts *Single-Session-Assistant* queries (-3.5) and *Single-Session Preference* queries (-6.7), indicat-

Table 4: Efficiency comparison on LONGMEMEVAL_S. We report accuracy, token cost (in thousands), and recall latency in seconds. Lower latency is better.

Method	ACC	Token Cost (k)					Recall Latency ↓ (P50 / P95, s)
		Summary In	Summary Out	Update In	Update Out	Total	
Baseline Methods							
A-MEM	62.60	214.66	42.82	1,157.52	190.81	1,605.81	5.12 / 11.89
Mem0	53.61	424.13	17.76	150.56	1,152.62	1,745.07	3.64 / 6.11
MemoryOS	44.80	2,302.35	304.18	350.02	35.19	2,991.74	1.63 / 3.95
TSM Variants							
TSM (Ours)	**74.80**	1,430.71	22.09	609.09	3.65	2,065.54	**1.57 / 2.39**
TSM (Sleep-Time Update)	–	1,334.06	12.72	–	–	1,959.53	–

ing that temporal misalignment affects downstream response generation even when the query is not explicitly time-sensitive.

Removing summaries while keeping temporal reranking ("w.o.summary") results in a decrease in overall accuracy (74.8% → 73.6%, −1.2), and notably hurts *Single-Session-Preference* (−10.0). This suggests that persona information can be beneficial for user-specific and preference-centric queries. It also significantly harms *Temporal* tasks (−4.5), implying that durative memories are crucial for capturing the narrative of past experiences. However, it improves performance on *Single-Session-User* (+1.5), *Knowledge-Update* (+1.3), and *Multi-Session* (+1.5), possibly due to the removal of distractions in certain categories.

Overall, both components contribute to performance, with temporal modeling offering the most consistent and substantial improvements.

5 Conclusions

In this paper, we presented TSM, a novel approach for memory construction and utilization that addresses key limitations of existing methods. By incorporating both semantic-time grounded and duration-aware management, TSM overcame the fragmentation caused by isolated, point-wise memory entries, ensuring that long-term, continuous user experiences are captured. The proposed duration-aware memory construction consolidates temporally continuous and semantically related information, enhancing contextual consistency and enabling more accurate retrieval of relevant memories. Additionally, the integration of semantic-time guided memory utilization improved retrieval by considering the temporal intent behind user queries. Extensive experiments on LONGMEMEVAL and LOCOMO datasets demonstrated that TSM significantly outperforms existing memory baselines, achieving notable improvements in QA accuracy, particularly in tasks requiring multi-session understanding and temporal reasoning. These results highlighted the importance of modeling semantic time and duration for effective, reliable long-term memory in LLM-based agents.

Limitations

While TSM demonstrates significant improvements in retrieval relevance and response quality, several limitations warrant discussion. First, TSM adopts a fixed temporal granularity (e.g., monthly intervals) for grouping durative summaries, which may not be optimal across all application domains. Adaptive granularity selection based on the temporal density of events could improve flexibility but is left for future work. Second, our work focuses on personalization applications. Extending our approach to other memory paradigms, such as procedural memory for agent learning and shared memory for multi-agent systems, remains important future work.

Acknowledgments

This work is partially funded by the National Key Research and Development Program of China under Grants No. 2024YFC3308200, the Strategic Priority Research Program of the CAS under Grants No. XDB0680102, the National Natural Science Foundation of China under grants 62306299, 62441229 and 62406308. We thank anonymous reviewers for their insightful comments and suggestions.

References

ByteDance. 2025. Deerflow: Deep exploration and efficient research framework. https://deerflow.tech/z.

Camel-AI. 2025. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation.

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

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025b. Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. 2025. LightMem: Lightweight and Efficient Memory-Augmented Generation. *arXiv preprint*. ArXiv:2510.18866 [cs] TLDR: Inspired by the Atkinson-Shiffrin model of human memory, LightMem organizes memory into three complementary stages, which strikes a balance between the performance and efficiency of memory systems.

Google. 2025. Gemini deep research — your personal research assistant. https://gemini.google/overview/deep-research/?hl=en-GB.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxiang Sun, Yutao Zhu, Hao Sun, Boci Peng, and 28 others. 2025. Memory in the Age of AI Agents. *arXiv preprint*. ArXiv:2512.13564 [cs].

Haoyu Huang, Yongfeng Huang, Junjie Yang, Zhenyu Pan, Yongqiang Chen, Kaili Ma, Hongzhi Chen, and James Cheng. 2025. Retrieval-Augmented Generation with Hierarchical Knowledge. *arXiv preprint*. ArXiv:2503.10150 [cs] TLDR: This paper introduces a new RAG approach, called HiRAG, which utilizes hierarchical knowledge to enhance the semantic understanding and structure capturing capabilities of RAG systems in the indexing and retrieval processes.

Alexis Huet, Zied Ben Houidi, and Dario Rossi. 2025. Episodic Memories Generation and Evaluation Benchmark for Large Language Models. *arXiv preprint*. ArXiv:2501.13121 [cs] TLDR: It is argued that integrating episodic memory capabilities into LLM is essential for advancing AI towards human-like cognition, increasing their potential to reason consistently and ground their output in real-world episodic events, hence avoiding confabulations.

David Kadavy. 2021. *Digital Zettelkasten: Principles, Methods, & Examples*. Kadavy, Inc. Google-Books-ID: o4gwEAAAQBAJ.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI Agent.

Sangyeop Kim, Yohan Lee, Sanghwa Kim, Hyunjong Kim, and Sungzoon Cho. 2025. Pre-Storage Reasoning for Episodic Memory: Shifting Inference Burden to Memory for Personalized Dialogue. *arXiv preprint*. ArXiv:2509.10852 [cs] TLDR: This work introduces PREMem (Pre-storage Reasoning for Episodic Memory), a novel approach that shifts complex reasoning processes from inference to memory construction, and creates enriched representations while reducing computational demands during interactions.

Taeyoon Kwon, Dongwook Choi, Sunghwan Kim, Hyojun Kim, Seungjun Moon, Beong-woo Kwak, Kuan-Hao Huang, and Jinyoung Yeo. 2025. Embodied Agents Meet Personalization: Exploring Memory Utilization for Personalized Assistance. *arXiv preprint*. ArXiv:2505.16348 [cs] TLDR: MEMENTO is presented, a personalized embodied agent evaluation framework designed to comprehensively assess memory utilization capabilities to provide personalized assistance, which consists of a two-stage memory evaluation process design that enables quantifying the impact of memory utilization on task performance.

LangChain. LangChain Blog.

Hao Li, Chenghao Yang, An Zhang, Yang Deng, Xiang Wang, and Tat-Seng Chua. 2025. Hello Again! LLM-powered Personalized Agent for Long-term Dialogue. *arXiv preprint*. ArXiv:2406.05925 [cs].

Xinbin Liang, Jinyu Xiang, Zhaoyang Yu, Jiayi Zhang, Sirui Hong, Sheng Fan, and Xiao Tang. 2025. Openmanus: An open-source framework for building general ai agents.

Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, Rongcheng Tu, Xiao Luo, Wei Ju, Zhiping Xiao, Yifan Wang, Meng Xiao, Chenwu Liu, Jingyang Yuan, Shichang Zhang, and 7 others. 2025. Large Language Model Agent: A Survey on Methodology, Applications and Challenges. *arXiv preprint*. ArXiv:2503.21460 [cs].

Christopher J. MacDonald, Kyle Q. Lepage, Uri T. Eden, and Howard Eichenbaum. 2011. Hippocampal "time cells" bridge the gap in memory for discontiguous events. *Neuron*, 71(4):737–749. TLDR: A robust hippocampal representation of sequence memories is reported, highlighted by "time cells" that encode successive moments during an empty temporal gap between the key events, while also encoding location and ongoing behavior.

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 [cs] TLDR: A machine-human pipeline is introduced to generate high-quality, very long-term dialogues by leveraging LLM-based agent architectures and grounding their dialogues on personas and temporal event graphs, and presents a comprehensive evaluation benchmark to measure long-term memory in models.

Andrea Matarazzo and Riccardo Torlone. 2025. A Survey on Large Language Models with some Insights on their Capabilities and Limitations. *arXiv preprint*. ArXiv:2501.04040 [cs].

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2025. Large Language Models: A Survey. *arXiv preprint*. ArXiv:2402.06196 [cs].

Jiayan Nan, Wenquan Ma, Wenlong Wu, and Yize Chen. 2025. Nemori: Self-Organizing Agent Memory Inspired by Cognitive Science. *arXiv preprint*. ArXiv:2508.03341 [cs] TLDR: Nemori is a novel self-organizing memory architecture inspired by human cognitive principles that significantly outperforms prior state-of-the-art systems, with its advantage being particularly pronounced in longer contexts.

OpenAI. 2024. GPT-4o mini: advancing cost-efficient intelligence.

Siru Ouyang, Jun Yan, I.-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. 2025. ReasoningBank: Scaling Agent Self-Evolving with Reasoning Memory. *arXiv preprint*. ArXiv:2509.25140 [cs] TLDR: Memory-driven experience scaling is established as a new scaling dimension, enabling agents to self-evolve with emergent behaviors naturally arise, and further introduces memory-aware test-time scaling (MaTTS), which accelerates and diversifies this learning process by scaling up the agent's interaction experience.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. MemGPT: Towards LLMs as Operating Systems. *arXiv preprint*. ArXiv:2310.08560 [cs] TLDR: This work introduces MemGPT (Memory-GPT), a system that intelligently manages different memory tiers in order to effectively provide extended context within the LLM's limited context window, and utilizes interrupts to manage control flow between itself and the user.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. *arXiv preprint*. ArXiv:2304.03442 [cs].

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025a. Zep: A Temporal Knowledge Graph Architecture for Agent Memory. *arXiv preprint*. ArXiv:2501.13956 [cs] TLDR: Zep is introduced, a novel memory layer service for AI agents that outperforms the current state-of-the-art system, MemGPT, in the Deep Memory Retrieval (DMR) benchmark and is validated through the more challenging LongMemEval benchmark, which better reflects enterprise use cases through complex temporal reasoning tasks.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025b. Zep: A Temporal Knowledge Graph Architecture for Agent Memory. *arXiv preprint*. ArXiv:2501.13956 [cs] TLDR: Zep is introduced, a novel memory layer service for AI agents that outperforms the current state-of-the-art system, MemGPT, in the Deep Memory Retrieval (DMR) benchmark and is validated through the more challenging LongMemEval benchmark, which better reflects enterprise use cases through complex temporal reasoning tasks.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Theodore Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas Griffiths. 2023. Cognitive architectures for language agents. *Transactions on Machine Learning Research*.

Haoran Sun and Shaoning Zeng. 2025. Hierarchical Memory for High-Efficiency Long-Term Reasoning in LLM Agents. *arXiv preprint*. ArXiv:2507.22925 [cs] TLDR: A Hierarchical Memory (H-MEM) architecture for LLM Agents is proposed that organizes and updates memory in a multi-level fashion based on the degree of semantic abstraction and consistently outperforms five baseline methods in long-term dialogue scenarios.

Zhen Tan, Jun Yan, I.-Hung Hsu, Rujun Han, Zifeng Wang, Long T. Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. 2025. In Prospect and Retrospect: Reflective Memory Management for Long-term Personalized Dialogue Agents. *arXiv preprint*. ArXiv:2503.08026 [cs] TLDR: Reflective Memory Management (RMM) is proposed, a novel mechanism for long-term dialogue agents, integrating forward- and backward-looking reflections: Prospective Reflection, which dynamically summarizes interactions across granularities-utterances, turns, and sessions-into a personalized memory bank for effective future retrieval, and Retrospective Reflection, which iteratively refines the retrieval in an online reinforcement learning (RL) manner based on LLMs'cited evidence.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025a. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. *arXiv preprint*. ArXiv:2410.10813 [cs] TLDR: This study introduces LongMemEval, a comprehensive benchmark designed to evaluate five core long-term memory abilities of chat assistants: information extraction, multi-session reasoning, temporal reasoning, knowledge updates, and abstention, and proposes several memory design optimizations including session decomposition for value granularity, fact-augmented key expansion for indexing, and time-aware query expansion for refining the search scope.

Tingyu Wu, Zhisheng Chen, Ziyan Weng, Shuhe Wang, Chenglong Li, Shuo Zhang, Sen Hu, Silin Wu, Qizhen Lan, Huacan Wang, and 1 others. 2026. Knowme-bench: Benchmarking person understanding for lifelong digital companions. *arXiv preprint arXiv:2601.04745*.

Yaxiong Wu, Sheng Liang, Chen Zhang, Yichao Wang, Yongyue Zhang, Huifeng Guo, Ruiming Tang, and Yong Liu. 2025b. From Human Memory to AI Memory: A Survey on Memory Mechanisms in the Era of LLMs. *arXiv preprint*. ArXiv:2504.15965 [cs].

Zhaofen Wu, Hanrong Zhang, Fulin Lin, Wujiang Xu, Xinran Xu, Yankai Chen, Henry Peng Zou, Shaowen Chen, Weizhi Zhang, Xue Liu, Philip S. Yu, and Hongwei Wang. GAM: Hierarchical Graph-based Agentic Memory for LLM Agents. *Preprint*, arXiv:2604.12285.

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

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Hinrich Schütze, Volker Tresp, and Yunpu Ma. 2025. Memory-R1: Enhancing Large Language Model Agents to Manage and Utilize Memories via Reinforcement Learning. *arXiv preprint*. ArXiv:2508.19828 [cs] TLDR: Memory-R1 is presented, a reinforcement learning (RL) framework that equips LLMs with the ability to actively manage and utilize external memory through two specialized agents: a Memory Manager that learns structured operations, including ADD, UPDATE, DELETE, and NOOP and an Answer Agent that pre-selects and reasons over relevant entries.

Chang Yang, Chuang Zhou, Yilin Xiao, Su Dong, Luyao Zhuang, Yujing Zhang, Zhu Wang, Zijin Hong, Zheng Yuan, Zhishang Xiang, and 1 others. 2026. Graph-based agent memory: Taxonomy, techniques, and applications. *arXiv preprint arXiv:2602.05665*.

Guibin Zhang, Muxin Fu, Guancheng Wan, Miao Yu, Kun Wang, and Shuicheng Yan. 2025a. G-Memory: Tracing Hierarchical Memory for Multi-Agent Systems. *arXiv preprint*. ArXiv:2506.07398 [cs] TLDR: G-Memory is introduced, a hierarchical, agentic memory system for MAS inspired by organizational memory theory, which manages the lengthy MAS interaction via a three-tier graph hierarchy: insight, query, and interaction graphs.

Guibin Zhang, Muxin Fu, and Shuicheng Yan. 2025b. MemGen: Weaving Generative Latent Memory for Self-Evolving Agents. *arXiv preprint*. ArXiv:2509.24704 [cs] TLDR: MemGen is proposed, a dynamic generative memory framework that equips agents with a human-esque cognitive faculty and spontaneously evolves distinct human-like memory faculties, including planning memory, procedural memory, and working memory, suggesting an emergent trajectory toward more naturalistic forms of machine cognition.

Wenyuan Zhang, Xinghua Zhang, Haiyang Yu, Shuaiyi Nie, Bingli Wu, Juwei Yue, Tingwen Liu, and Yongbin Li. 2026. Expseek: Self-triggered experience seeking for web agents. *Preprint*, arXiv:2601.08605.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2023. MemoryBank: Enhancing Large Language Models with Long-Term Memory. *arXiv preprint*. ArXiv:2305.10250 [cs] TLDR: MemoryBank incorporates a memory updating mechanism, inspired by the Ebbinghaus Forgetting Curve theory, that permits the AI to forget and reinforce memory based on time elapsed and the relative significance of the memory, thereby offering a more human-like memory mechanism and enriched user experience.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. 2025. MEM1: Learning to Synergize Memory and Reasoning for Efficient Long-Horizon Agents. *arXiv preprint*. ArXiv:2506.15841 [cs] TLDR: MEM1, an end-to-end reinforcement learning framework that enables agents to operate with constant memory across long multi-turn tasks, is introduced, an end-to-end reinforcement learning framework that enables agents to operate with constant memory across long multi-turn tasks.

A Appendix

A.1 Implementation Details

A.1.1 Parameter Setup

We use the following hyper-parameters for all experiments:

- **LLM:** GPT-4o-mini and Qwen3-30B-A3B-Instruct-2507 are used for all stages. During the memory construction stage, we only use the user message for efficiency. During the utilization stage, the generation parameters are:
  - Temperature: 0.0

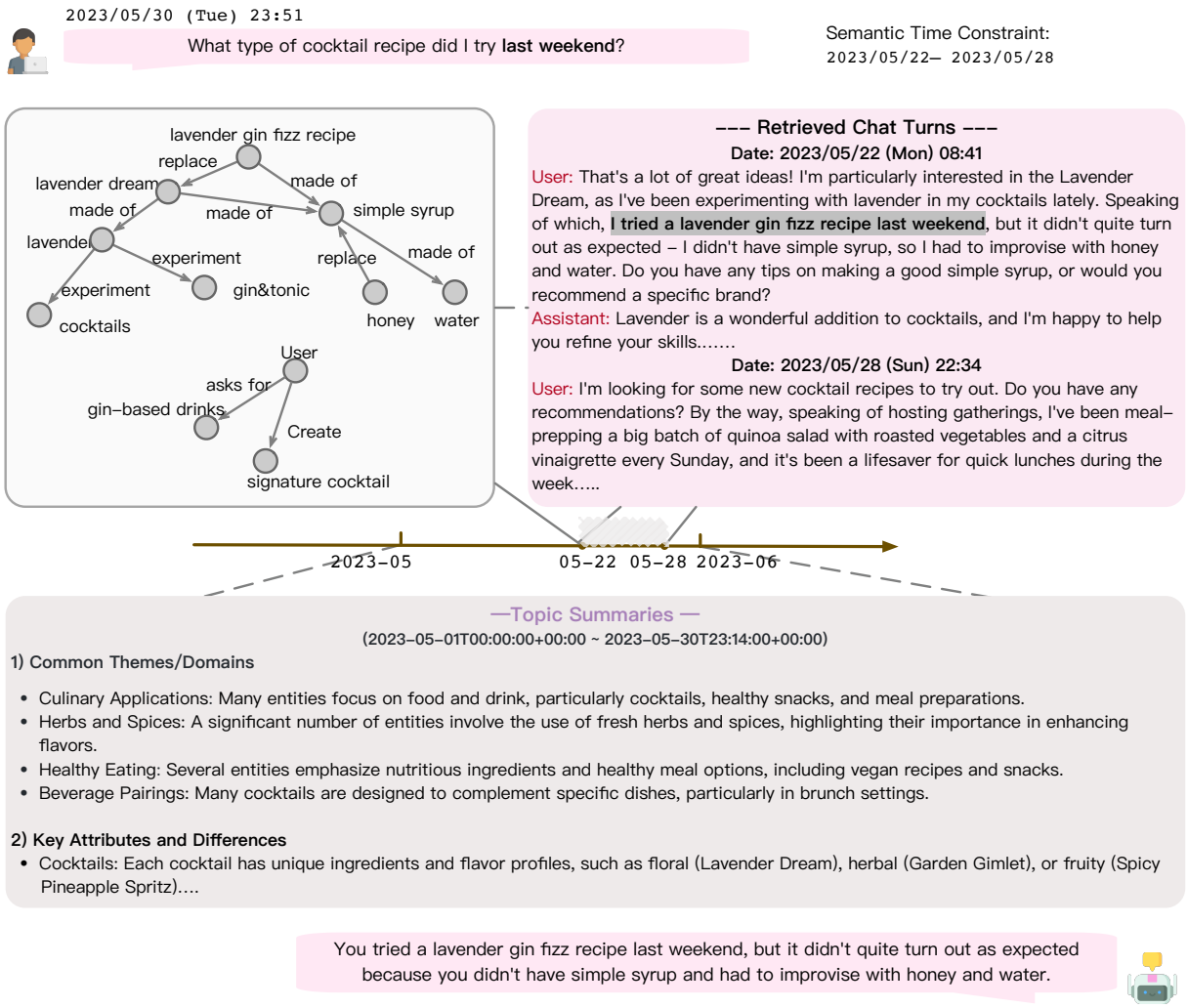

Figure 3: Case Study of TSM.

Table 5: Performance comparison of different embedding models with GPT-4o-mini backbone on LoCoMo dataset.

Embedding Model	ACC	Temporal	Multi-Hop	Open-Domain	Single-Hop
`text-embedding-v4`	73.01	68.51	60.80	**59.55**	80.37
`text-embedding3-small`	**76.69**	**71.03**	**66.67**	58.33	**84.30**

  - Max tokens: 8192

- **Retriever:** For the embedding model, we use `text-embedding-3-small` from OpenAI. Additionally, we performed experiments with `text-embedding-v4` from Qwen. Experimental results are listed in Table 5. We use the following configuration:

  - Top-K: 25

A.1.2 Hardware

Experiments are conducted on a machine equipped with 8 Nvidia A100 GPUs, each with 80 GB of RAM. The total available system memory is 256 GB.

A.2 Case Study

Figure 3 illustrates a concrete memory Utilization of TSM. In this example, the user asks about a cocktail recipe. Given the query time (2023-05-30) and the temporal expression "last weekend," SpaCy parses the corresponding semantic time range as 2023-05-22 to 2023-05-28. Based on this temporal constraint, TSM first queries the TKG to retrieve all facts whose valid time falls within the identified interval. Then, TSM performs dense retrieval over both raw dialogue and summaries. Only summaries whose temporal scope satisfies the query's time constraint are considered. The retrieved facts from the TKG are then used as contextual signals to rerank the candidate chunks, promoting those that are semantically aligned with the time constraint. As a result, the chunk containing the correct cocktail recipe is ranked at the top.

During the graph retrieval, although several related facts are found, none of them can directly answer the user's question. This highlights a key limitation of using TKG facts alone as ground-truth memory: while they capture structured and time-aware information, they contain insufficient, pointwise, and instant knowledge.

Overall, TSM effectively combines temporal reasoning and semantic retrieval to produce accurate and temporally aligned responses, demonstrating its advantage over methods that rely solely on timestamped facts or unfiltered dense retrieval.

A.3 Datasets and Baseliens

Datasets. The LongMemEval dataset is a comprehensive, challenging, and scalable benchmark for testing the long-term memory of chat assistants. Two standard test sets are created for 500 questions: LongMemEval_S with each question's chat history has roughly 115k tokens (30-40 sessions) and LongMemEval_M: each question's chat history has roughly 500 sessions (1.5M tokens). In our work, we adopt the LongMemEval-S version due to its balance between dialogue length and computational feasibility.

The LoCoMo benchmark targets the evaluation of long-range conversational memory. It features extremely long dialogues, with each conversation spanning roughly 300 turns and around 9K tokens on average. Note that questions in the LoCoMo dataset do not contain explicit query timestamps. We use the session start time as the reference timestamp when extracting temporal constraints from queries.

Baselines. (1) LangMem (LangChain) is the Langchain's long-term memory module.
(2) A-MEM (Xu et al., 2025) system dynamically structures memories through notes. Each note has attributes like keywords, contextual descriptions, and tags generated by the LLM. Retrieval from memory is conducted through semantic similarity.
(3) MemoryOS (Kang et al.) organizes conversational memory in an OS-inspired hierarchy, structuring interactions into short-term, mid-term, and long-term layers via paging and heat-based updating.
(4) Mem0 (Chhikara et al., 2025a) extracts memories from dialogue turns through a combination of global summaries and recent context, maintaining them via LLM-guided operations. Mem0^g further propose an enhanced variant that leverages graph-based memory representations to capture complex relational structures among conversational elements.

(5) Zep (Rasmussen et al., 2025a) is a temporal knowledge graph architecture that organizes data into episodic, semantic, and community subgraphs to capture dynamic, time-sensitive relationships. For comparison, we include the baseline results from their original papers and LightMem (Fang et al., 2025). Due to computational resource constraints, all experiments are conducted with a single run using a fixed setting.

A.4 LLM-as-Judge prompts

Standard Tasks (Single-session-user/assistant Multi-session) for LongMemEval Dataset

I will give you a question, a correct answer, and a response from a model. Please answer yes if the response contains the correct answer. Otherwise, answer no. If the response is equivalent to the correct answer or contains all the intermediate steps to get the correct answer, you should also answer yes. If the response only contains a subset of the information required by the answer, answer no.
Question: {question}
Correct Answer: {answer}
Model Response: {response}
Is the model response correct? Answer yes or no only.

Temporal Reasoning Tasks for LongMemEval Dataset

I will give you a question, a correct answer, and a response from a model. Please answer yes if the response contains the correct answer. Otherwise, answer no. If the response is equivalent to the correct answer or contains all the intermediate steps to get the correct answer, you should also answer yes. If the response only contains a subset of the information required by the answer, answer no. In addition, do not penalize off-by-one errors for the number of days. If the question asks for the number of days/weeks/months, etc., and the model makes off-by-one errors (e.g., predicting 19 days when the answer is 18), the model's response is still correct.
Question: {question}
Correct Answer: {answer}
Model Response: {response}
Is the model response correct? Answer yes or no only.

Knowledge Update Tasks for LongMemEval Dataset

I will give you a question, a correct answer, and a response from a model. Please answer yes if the response contains the correct answer. Otherwise, answer no. If the response contains some previous information along with an updated answer, the response should be considered as correct as long as the updated answer is the required answer.
Question: {question}
Correct Answer: {answer}
Model Response: {response}
Is the model response correct? Answer yes or no only.

Single-session Preference Tasks for LongMemEval Dataset

I will give you a question, a rubric for desired personalized response, and a response from a model. Please answer yes if the response satisfies the desired response. Otherwise, answer no. The model does not need to reflect all the points in the rubric. The response is correct as long as it recalls and utilizes the user's personal information correctly.
Question: {question}
Rubric: {answer}
Model Response: {response}
Is the model response correct? Answer yes or no only.

Abstention Tasks for LongMemEval Dataset

I will give you an unanswerable question, an explanation, and a response from a model. Please answer yes if the model correctly identifies the question as unanswerable. The model could say that the information is incomplete, or some other information is given but the asked information is not.
Question: {question}
Explanation: {answer}
Model Response: {response}
Does the model correctly identify the question as unanswerable? Answer yes or no only.

LoCoMo Dataset

Your task is to label an answer to a question as 'CORRECT' or 'WRONG'. You will be given the following data:
(1) a question (posed by one user to another user),
(2) a 'gold' (ground truth) answer,
(3) a generated answer
which you will score as CORRECT/WRONG.

The point of the question is to ask about something one user should know about the other user based on their prior conversations.
The gold answer will usually be a concise and short answer that includes the referenced topic, for example:
Question: Do you remember what I got the last time I went to Hawaii?
Gold answer: A shell necklace
The generated answer might be much longer, but you should be generous with your grading - as long as it touches on the same topic as the gold answer, it should be counted as CORRECT.

For time related questions, the gold answer will be a specific date, month, year, etc. The generated answer might be much longer or use relative time references (like "last Tuesday" or "next month"), but you should be generous with your grading - as long as it refers to the same date or time period as the gold answer, it should be counted as CORRECT. Even if the format differs (e.g., "May 7th" vs "7 May"), consider it CORRECT if it's the same date.
Now it's time for the real question:

Question: {question}
Gold answer: {gold_answer}
Generated answer: {generated_answer}

First, provide a short (one sentence) explanation of your reasoning, then finish with CORRECT or WRONG. Do NOT include both CORRECT and WRONG in your response, or it will break the evaluation script.

Just return the label CORRECT or WRONG in a json format with the key as "label".