TEST-TIME MEMORY CONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) struggle to answer questions over long interaction histories, where relevant evidence may be scattered across distant sessions. Existing memory systems typically construct and maintain sophisticated memory representations offline, before user queries are observed, incurring substantial overhead and limiting their ability to adapt memory construction to query-specific information needs. In this paper, we propose T^{2}Mem, a **T**est-**T**ime **Mem**ory construction framework that defers memory construction to inference and progressively builds memory along the reasoning process. Specifically, T^{2}Mem extracts lightweight episode-level metadata offline to capture key events and temporal associations as retrieval cues, while retaining links to the original interactions to facilitate the reconstruction of dependencies across episodes. At test time, the LLM adaptively retrieves and organizes metadata and raw interactions, progressively constructing query-specific memory as reasoning unfolds. Experiments on LoCoMo and LongMemEval show the effectiveness of T^{2}Mem by consistently outperforming strong baselines across different backbone LLMs while substantially reducing the overhead of memory construction. Further analysis demonstrates that deferring memory construction to inference enables the memory construction process to dynamically adapt to the information needs of individual queries. All code and datasets will be available via GitHub.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated strong emergent capabilities across a wide range of tasks (Yao et al., 2023; Li et al., 2025; Xin et al., 2026b; Liu et al., 2026). However, as user interactions grow longer, LLMs still struggle to reason over extended interaction histories due to limited context windows and the difficulty of effectively locating relevant information within long contexts (Hu et al., 2025; Liu et al., 2024; Hsieh et al., 2024). A straightforward solution is Retrieval-Augmented Generation (RAG), which segments historical interactions into chunks and retrieves relevant ones at inference time (Lewis et al., 2020; Edge et al., 2024; Li et al., 2025; 2026b). However, chunk-level retrieval often treats historical interactions as isolated units, potentially disrupting temporal context and cross-turn dependencies among related interactions. This fragmentation makes it difficult for LLMs to fully exploit the interconnected information embedded in long interaction histories (Xin et al., 2026a; Zhang et al., 2025).

To address this limitation, recent memory systems (Chhikara et al., 2025; Kang et al., 2025; Fang et al., 2026) organize long interaction histories into manageable memory units. These methods introduce dedicated mechanisms to filter, summarize, and consolidate historical interactions into persistent memory representations, enabling efficient access in future interactions. However, constructing and maintaining such sophisticated memory representations can incur substantial overhead. In particular, offline memory construction must anticipate diverse future queries and preserve sufficiently comprehensive information in each memory unit, resulting in considerable preprocessing and maintenance costs that can hinder the scalability and practical deployment of memory systems (Peng et al., 2026). Recent efforts have therefore explored deferring memory construction to test time, reducing the reliance on costly preconstructed memory representations (Yu et al., 2026; Yin & Tang, 2026). Nevertheless, these approaches typically construct memory from an initially retrieved candidate set, making it difficult to acquire additional evidence that is not retrieved initially but becomes necessary as the memory construction process unfolds.

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

Recent advances in test-time scaling have demonstrated the potential of leveraging additional inference-time computation to elicit stronger reasoning capabilities from LLMs. By allocating more computation at inference time, LLMs can iteratively explore, verify, and revise intermediate solutions (Snell et al., 2025; OpenAI et al., 2024; Guo et al., 2025). This raises a natural question: *Can memory construction similarly benefit from inference-time reasoning, allowing an LLM to dynamically determine what historical information to activate and organize as it reasons?* We answer this question by treating memory not as a fixed representation constructed offline, but as an evolving state that is progressively constructed during inference.

Based on this perspective, we propose T^{2}Mem, a **T**est-**T**ime **Mem**ory construction framework that dynamically constructs memory during inference. T^{2}Mem draws inspiration from associative recall, where relevant memories can be progressively activated as retrieval cues emerge (Teyler & Rudy, 2007; Horner et al., 2015). Instead of constructing sophisticated memory representations offline, T^{2}Mem first extracts lightweight episode-level metadata that captures salient events and temporal associations, while retaining backlinks to the corresponding raw interactions. The metadata provides a compact structure for associating related episodes and locating their underlying evidence. During inference, the LLM progressively reasons over the query and uses this evolving reasoning process to navigate the metadata and selectively activate corresponding raw interactions. In this way, memory construction and reasoning are interleaved: each newly activated piece of evidence can inform subsequent reasoning and, in turn, guide the construction of a more relevant memory state. The final memory is therefore not predetermined before inference, but emerges progressively from the interaction between reasoning and memory activation.

In our experiments, we evaluate T^{2}Mem on LoCoMo and LongMemEval with two backbone LLMs. Across both benchmarks, T^{2}Mem consistently outperforms strong memory and retrieval baselines under comparable total token budgets, demonstrating the effectiveness of dynamically constructing memory at test time. Further analysis reveals that deferring memory construction to inference allows T^{2}Mem to adapt the construction process to the specific information needs of each query. This advantage is particularly evident for queries that require deeper memory construction, such as multi-hop and temporal questions, where relevant evidence is distributed across multiple episodes and cannot be fully identified through a single initial retrieval step. Moreover, by leveraging episode-level metadata, T^{2}Mem enables the LLM to establish dependencies across episodes and trace these dependencies back to relevant source interactions during reasoning, facilitating more effective memory construction without the substantial overhead of maintaining sophisticated memory units.

2 Related Work

Large Language Models (LLMs) struggle to reason over long-horizon interaction histories due to limited context windows (Liu et al., 2024; Hsieh et al., 2024). Retrieval-Augmented Generation (RAG) methods address this limitation by segmenting interaction histories into chunks and retrieving query-relevant ones as context for answering user queries (Lewis et al., 2020; Edge et al., 2024). However, such chunk-based retrieval often breaks the contextual and temporal dependencies inherent in interaction histories (Xin et al., 2026a). Recent memory systems therefore introduce dedicated memory management mechanisms to organize, retain, and update historical interactions. For example, MemGPT (Packer et al., 2023) and MemoryOS (Kang et al., 2025) are inspired by operating systems and maintain memory states by continuously incorporating relevant information into existing memory units. MemoryBank (Zhong et al., 2024) draws inspiration from human memory and incorporates the Ebbinghaus forgetting curve to regulate the retention and updating of memory units. Despite their effectiveness, these approaches primarily focus on memory maintenance, while the structure and representation of memory units, which can directly affect how effectively memories are retained, updated, and accessed, remain less explored.

To construct more effective memory representations, existing methods typically segment, compress, and consolidate interaction histories using different strategies. Specifically, HiMem (Zhang et al., 2026) segments interaction histories along event boundaries to form coherent episodic memories, while LightMem (Fang et al., 2026) compresses and consolidates interactions by topic to preserve salient information. Subsequent approaches further organize these compact memories into higher-level semantic (Hu et al., 2026b) or temporal (Li et al., 2026a) structures, allowing evidence distributed across interactions to be accessed at different levels of abstraction. To further enhance

the traceability of memory representations, AnchorMem (Shen et al., 2026) links memory units to their original interactions, allowing fine-grained source information to be retrieved when needed. However, these approaches often incur substantial overhead in constructing and maintaining sophisticated memory representations, which can hinder the practical deployment and scalability of memory systems (Peng et al., 2026).

To alleviate the reliance of memory effectiveness on predefined memory representations, recent work has shifted toward query-time memory construction (Yu et al., 2026; Yin & Tang, 2026). These methods typically retrieve a broad set of query-relevant facts and reorganize them into memory. While this paradigm enables more comprehensive memory construction, its retrieval process remains largely rigid and one-shot, focusing on retrieving more relevant information rather than actively identifying what information is actually needed to construct an effective memory. Such a passive retrieval process may therefore fail to acquire missing evidence that only becomes apparent from the partially constructed memory (Yao et al., 2023; Li et al., 2025). To enable LLMs to actively acquire information according to their evolving needs, T^{2}Mem treats memory construction as a dynamic reasoning process at test time. Inspired by the test-time scaling of LLMs (OpenAI et al., 2024; Guo et al., 2025; Snell et al., 2025), T^{2}Mem allows the evolving memory state to actively determine what evidence should be retrieved and integrated next. This closed-loop process enables the model to seek information genuinely needed for memory construction, better distinguish between potentially conflicting or ambiguous facts, and progressively refine memory as reasoning unfolds.

3 METHODOLOGY

In this section, we introduce T^{2}Mem, a *test-time memory construction* framework for long-term conversational question answering. Unlike conventional approaches that construct a query-agnostic memory before inference, T^{2}Mem dynamically constructs memory at test time by progressively uncovering and organizing relevant information from the raw interaction history. We first revisit existing memory systems and show how T^{2}Mem decouples memory construction from the offline preprocessing stage, enabling memory to be constructed dynamically during inference (Sec. 3.1). We then present the architecture of T^{2}Mem, which leverages the reasoning capabilities of LLMs to progressively incorporate relevant information to construct tailored memory (Sec. 3.2).

3.1 DISENTANGLING MEMORY ORCHESTRATION FROM OFFLINE TO TEST TIME

Given a long-term interaction history $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$, a memory system organizes information scattered across interaction sessions into manageable memory units $\mathcal{M} = \{m_1, m_2, \ldots, m_N\}$. These memory units focus on preserving factual knowledge, user preferences, and cross-session dependencies by resolving conflicts in the interactions and incorporating potentially useful information. Existing memory systems typically construct such memory representations offline, before future queries are presented to the LLM. In contrast, T^{2}Mem defers memory orchestration to test time, using the given query to incorporate acquired information into the reasoning trajectory, thereby organizing relevant knowledge dynamically during inference.

Offline Memory Construction. Existing memory systems first construct a query-agnostic memory collection $\mathcal{M}$ from the raw interaction history $\mathcal{S}$ by extracting, summarizing, and organizing potentially useful information:

$$\mathcal{M} = \pi_{\text{mem}}(s_1, s_2, \ldots, s_n), \tag{1}$$

where π_{mem} denotes the memory construction policy, which may repeatedly invoke LLMs to extract, update, and delete information as new interactions arrive, while also maintaining sophisticated memory representations such as hierarchical or graph-based structures. Moreover, since such construction is query-agnostic, the memory system must organize information without knowing which parts will be relevant to future queries, often incurring substantial construction and maintenance overhead (Peng et al., 2026). Given a user query q, the system retrieves a subset of the preconstructed memory units and uses them to generate the answer:

$$y = \text{LLM}(q, \text{Retriever}(q, \mathcal{M})), \tag{2}$$

where $\text{Retriever}(\cdot)$ denotes the memory retrieval procedure. By extracting and organizing information in advance, these methods maintain persistent memory representations that can be directly retrieved at inference time (Kang et al., 2025; Hu et al., 2026a).

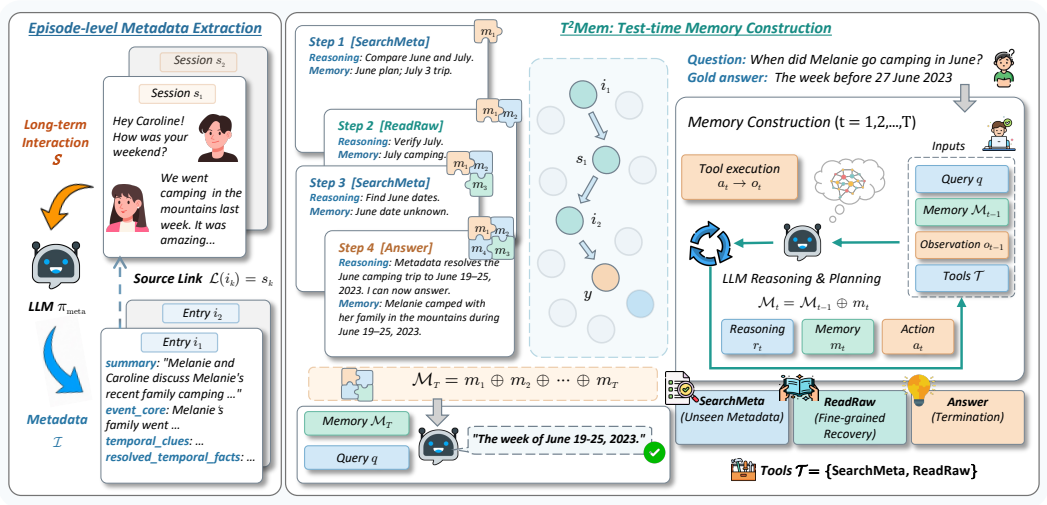


Figure 1: Overall Architecture of Our Proposed T^{2}Mem Framework.

Test-Time Memory Evolving. Rather than constructing sophisticated memory representations offline, T^{2}Mem mimics human associative recall by progressively activating relevant episodes as reasoning unfolds, allowing the LLM to orchestrate memory at test time (Horner et al., 2015). To represent episodic experiences (Fang et al., 2026), T^{2}Mem first extracts episode-level metadata that captures the dependencies among key events and temporal associations within each episode, using concise episode descriptions as retrieval cues. Given a query, the LLM then progressively uncovers query-relevant information by following these cues and locating the corresponding raw dialogue through the extracted metadata.

Specifically, T^{2}Mem employs an LLM π_{meta} to extract lightweight, episode-level metadata $\mathcal{I}$ from the dialog history $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$:

$$\mathcal{I} = \pi_{\text{meta}}(s_1, s_2, \ldots, s_n), \tag{3}$$

where $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$ contains extracted metadata entries for dialog sessions. Each i_k organizes the key events, facts, entities, and temporal cues from session s_k into a semi-structured representation, while maintaining a source link $\mathcal{L}(i_k) = s_k$ for traceability (Sun et al., 2026; Shen et al., 2026). These metadata entries serve as compact retrieval cues for associating related episodes at test time (Eq. 4), allowing T^{2}Mem to identify cross-episode dependencies through the metadata and trace the retrieved cues back to their corresponding source sessions via $\mathcal{L}(i_k)$.

Subsequently, given a query q, T^{2}Mem dynamically constructs memory by accessing the metadata $\mathcal{I}$ and the original interaction history $\mathcal{S}$ through a set of information-access tools $\mathcal{T}$:

$$y = \text{LLM}\big(q, \mathcal{I}, \mathcal{S}, \mathcal{T}\big). \tag{4}$$

Rather than relying on rigid, one-shot retrieval, T^{2}Mem invokes the tools in $\mathcal{T}$ during inference to progressively acquire relevant information in response to its evolving reasoning needs. The retrieved information is incorporated into the ongoing reasoning process, enabling the model to iteratively construct memory as reasoning unfolds. Thus, T^{2}Mem decouples memory construction from costly offline memory orchestration, instead deferring memory construction to test time, where it can be dynamically tailored to the query and the evolving reasoning process.

3.2 Dynamic Memory Construction During LLM Reasoning

As illustrated in Figure 1, we detail how T^{2}Mem dynamically constructs query-specific memory at test time (Eq. 4) by progressively acquiring, distilling, and organizing relevant information during reasoning. At each step, the LLM reasons over the accumulated memory and the current observation, distills query-relevant information into the evolving memory state, and determines what additional information is needed. Based on this information need, it either terminates the process or selects an

appropriate tool from $\mathcal{T}$ to access episode-level metadata or fine-grained raw information, thereby supporting subsequent memory construction.

We first formulate this process as an iterative reasoning trajectory τ consisting of T memory construction steps:

$$\tau = \big((r_1, m_1, a_1), (r_2, m_2, a_2), \ldots, (r_T, m_T, a_T)\big), \tag{5}$$

where r_t, m_t, and a_t denote the reasoning trace, newly constructed memory cache, and selected action at step t, respectively. The accumulated memory caches $\mathcal{M}_t$ generated throughout the trajectory are progressively concatenated to form the final query-specific memory $\mathcal{M}_T$:

$$\mathcal{M}_T = m_1 \oplus m_2 \oplus \cdots \oplus m_T, \tag{6}$$

where $\oplus$ denotes the concatenation operation. The LLM then generates the final answer conditioned on the user query q and the resulting memory $\mathcal{M}_T$:

$$y = \mathrm{LLM}(q, \mathcal{M}_T). \tag{7}$$

Dynamic Memory Construction. We next describe how each step produces (r_t, m_t, a_t) and updates the memory state. At step t, the LLM takes the query q, the accumulated memory $\mathcal{M}_{t-1}$, the latest observation o_{t-1}, and the available tool set $\mathcal{T}$ as input, and jointly produces a reasoning trace r_t, a newly constructed memory cache m_t, and the next action a_t:

$$(r_t, m_t, a_t) = \mathrm{LLM}(q, \mathcal{M}_{t-1}, o_{t-1}, \mathcal{T}), \tag{8}$$

where the memory is initialized as $\mathcal{M}_0 = \varnothing$. The initial observation is obtained by retrieving the top-k metadata entries in $\mathcal{I}$ most relevant to q:

$$o_0 = \text{Top-}\,k(q, \mathcal{I}), \tag{9}$$

where Top- k selects the k metadata entries in $\mathcal{I}$ that are most relevant to the query q. The reasoning trace r_t guides the LLM in distilling query-relevant evidence from o_{t-1} into the newly constructed memory cache m_t. The memory is then updated by appending m_t to the accumulated memory:

$$\mathcal{M}_t = \mathcal{M}_{t-1} \oplus m_t, \tag{10}$$

thereby completing the memory update at step t. The process terminates when the current memory state satisfies the information need of the given query or when the maximum number of construction steps $T_{\max}$ is reached. Let $T \le T_{\max}$ denote the actual number of executed steps. Finally, the accumulated memory $\mathcal{M}_T$ is used to help the LLM generate the final answer according to Eq. 7.

Adaptive Searching During LLM Reasoning. After constructing m_t, the LLM executes the selected action a_t, either terminating the process with `Answer` or invoking an information-access tool to obtain the next observation o_t. Specifically, we define the tool set $\mathcal{T}$ as:

$$\mathcal{T} = \{\texttt{SearchMeta}(\cdot), \texttt{ReadRaw}(\cdot)\}, \tag{11}$$

where $\texttt{SearchMeta}(\cdot)$ retrieves additional episode-level metadata, while $\texttt{ReadRaw}(\cdot)$ follows the source link $\mathcal{L}(i_k) = s_k$ associated with the observed metadata entry i_k to recover the corresponding raw session. To avoid repeatedly reading the same metadata entry, we maintain a query-specific set of previously observed entries, $\mathcal{I}^{\text{seen}} \subseteq \mathcal{I}$, initialized with the metadata entries contained in o_0.

When the current evidence is insufficient and additional events or temporal cues are required, the LLM selects $\texttt{SearchMeta}(\cdot)$ to acquire previously unseen metadata. The tool excludes entries already contained in $\mathcal{I}^{\text{seen}}$ and retrieves the highest-ranked remaining entries:

$$\texttt{SearchMeta}(q) = \text{Top-}\,k\left(q, \mathcal{I} \setminus \mathcal{I}^{\text{seen}}\right). \tag{12}$$

After each call to $\texttt{SearchMeta}(\cdot)$, the returned entries are added to $\mathcal{I}^{\text{seen}}$, preventing redundant reading. When an observed metadata entry i_k points to a relevant session but does not provide sufficient detail, the LLM can instead select $\texttt{ReadRaw}(k)$ to access the corresponding raw session:

$$\texttt{ReadRaw}(k) = \mathcal{L}(i_k) = s_k, \tag{13}$$

where $i_k \in \mathcal{I}^{\text{seen}}$. This operation enables the LLM to recover fine-grained evidence from the corresponding raw interaction when the lightweight metadata is insufficient, thereby allowing the memory construction process to adaptively incorporate fine-grained evidence into the evolving memory.

Table 1: Overall Performance of T^{2}Mem and Baseline Methods on the LoCoMo Dataset. The **best** and second-best results are highlighted.

Method	# Token ($\times 10^3$)	Multi-Hop	Temporal	Open-Domain	Single-Hop	Avg.
Qwen3.5-27B						
Naive RAG (2020)	1,012	32.98	60.12	38.54	64.68	56.30
Search-o1 (2025)	2,380	49.29	61.68	53.12	60.64	58.31
MemoryOS (2025)	6,460	51.77	41.12	42.71	72.77	60.45
LightMem (2026)	6,178	58.51	70.72	40.62	74.55	68.70
EverMemOS (2026a)	12,955	51.77	66.36	**58.33**	69.20	64.74
HiMem (2026)	11,056	59.03	68.37	49.81	71.17	65.34
xMemory (2026b)	8,229	57.09	71.96	47.92	72.65	68.12
AnchorMem (2026)	10,940	47.16	70.72	42.71	75.83	67.45
LazyMem (2026)	11,253	51.77	62.50	36.96	**82.28**	69.84
T^{2}Mem (Ours)	8,765	**59.22**	**75.70**	50.00	76.81	**71.69**
Gemma4-31B-Instruct						
Naive RAG (2020)	1,025	32.62	59.50	31.25	63.50	55.00
Search-o1 (2025)	1,891	39.36	57.63	44.79	58.03	53.70
MemoryOS (2025)	7,357	56.03	64.17	50.00	75.51	65.84
LightMem (2026)	6,326	**62.06**	**77.26**	37.50	72.06	69.16
EverMemOS (2026a)	11,063	52.48	65.42	50.00	66.47	62.66
HiMem (2026)	10,289	47.87	66.36	28.12	63.81	59.19
xMemory (2026b)	8,190	55.32	71.34	45.83	71.94	67.14
AnchorMem (2026)	8,949	48.94	76.01	40.62	74.52	68.03
LazyMem (2026)	10,624	52.48	62.50	36.96	**83.23**	70.49
T^{2}Mem (Ours)	7,274	56.74	**77.26**	**52.08**	76.10	**71.30**

4 Experimental Methodology

In this section, we describe the experimental setup for evaluating T^{2}Mem. We first introduce the datasets, baselines, and evaluation metrics, and then present the implementation details.

Datasets. We evaluate T^{2}Mem on two widely adopted benchmarks for long-term memory, LoCoMo (Maharana et al., 2024) and LongMemEval (Wu et al., 2025). These benchmarks assess the ability of memory systems to retain, retrieve, and utilize information from long-term interaction histories. Further details are provided in Appendix A.2.

Baselines. We compare T^{2}Mem with several representative baselines spanning three categories: retrieval-based methods, memory systems that construct structured memory representations offline, and recent test-time approaches that defer memory construction to inference.

For retrieval-based methods, we evaluate Naive RAG (Lewis et al., 2020) and Search-o1 (Li et al., 2025), which directly retrieve relevant dialogue segments to support LLM-based answering. We then compare T^{2}Mem with representative memory systems. MemoryOS (Kang et al., 2025) draws inspiration from operating systems and continuously incorporates relevant information into existing memory units. LightMem (Fang et al., 2026) and HiMem (Zhang et al., 2026) construct compact memory units through topic-based consolidation and event-level organization, respectively. EverMemOS (Hu et al., 2026a) and xMemory (Hu et al., 2026b) further organize memory into higher-level semantic structures to facilitate subsequent retrieval. AnchorMem (Shen et al., 2026) preserves links between memory units and their source interactions, enabling the recovery of fine-grained details when needed. Finally, we adopt LazyMem (Yu et al., 2026) as a test-time baseline, which follows a retrieve-then-construct paradigm to build memory from a broad set of retrieved candidates during inference.

Evaluation Metrics. Following prior work (Fang et al., 2026; Xin et al., 2026a), we use accuracy as the primary evaluation metric and employ an LLM-based judge to determine whether each response is consistent with the ground-truth answer (Zheng et al., 2023). For a fair comparison, all methods are evaluated using Qwen3.5-Plus (Qwen Team, 2026) as the judge model. The evaluation prompts are provided in Appendix A.8.

Implementation Details. In our experiments, we use Qwen3.5-27B (Qwen Team, 2026) and Gemma4-31B-Instruct (Team et al., 2026) as the backbone models for all compared methods, and

Table 2: Overall Performance of T^{2}Mem and Baseline Methods on the LongMemEval Dataset.

Method	Single User	Single Assistant	Single Preference	Multi Session	Knowledge Update	Temporal Reasoning	Avg.
Qwen3.5-27B							
Naive RAG (2020)	92.86	87.50	36.67	69.92	79.49	58.65	71.60
Search-o1 (2025)	88.57	89.29	40.00	57.14	75.64	59.40	67.60
MemoryOS (2025)	90.00	82.14	36.67	60.15	69.23	48.87	63.80
LightMem (2026)	91.43	25.00	76.67	**78.20**	87.18	**85.71**	77.40
EverMemOS (2026a)	72.68	80.36	33.33	55.64	70.51	52.63	61.00
HiMem (2026)	75.71	50.00	33.33	54.14	70.51	56.39	58.60
xMemory (2026b)	94.29	64.29	46.67	60.90	83.33	51.65	68.80
AnchorMem (2026)	**95.71**	**98.21**	40.87	57.89	82.05	44.36	64.40
LazyMem (2026)	92.86	91.07	63.33	52.63	**94.87**	63.16	72.60
T^{2}Mem (Ours)	92.86	**98.21**	**83.33**	70.68	92.31	84.96	**84.80**
Gemma4-31B-Instruct							
Naive RAG (2020)	**97.14**	85.71	56.67	73.68	93.59	63.16	77.60
Search-o1 (2025)	94.29	94.64	33.33	62.41	73.08	62.41	70.40
MemoryOS (2025)	92.86	82.14	30.00	69.17	79.49	60.15	70.80
LightMem (2026)	92.86	10.71	**90.00**	**79.70**	87.18	**87.22**	77.60
EverMemOS (2026a)	88.57	78.57	60.00	70.68	88.46	62.41	74.00
HiMem (2026)	82.86	55.36	40.00	60.15	76.92	62.41	64.80
xMemory (2026b)	94.29	66.07	50.00	60.90	83.33	61.65	69.20
AnchorMem (2026)	95.71	98.21	36.67	69.17	**94.87**	63.91	76.80
LazyMem (2026)	95.71	96.43	73.33	55.64	91.03	69.92	76.20
T^{2}Mem (Ours)	94.29	**100.00**	83.33	69.92	92.31	85.71	**85.20**

BGE-M3 (Chen et al., 2024) as the embedding model for metadata retrieval. For T^{2}Mem, we set $k = 5$ for the initial retrieval from $\mathcal{I}$ to obtain o_0. In each subsequent `SearchMeta` call, we set $k = 1$ to retrieve the highest-ranked entry from $\mathcal{I} \setminus \mathcal{I}^{\text{seen}}$. We set the maximum number of memory construction steps to $T_{\max} = 5$ in the main experiments. The prompt templates used by T^{2}Mem are provided in Appendix A.8, while additional implementation details are reported in Appendix A.3.

5 Experimental Results

In this section, we first report the overall performance of T^{2}Mem and conduct ablation studies to show the effectiveness of different components. Then, we evaluate the quality of the memory constructed by T^{2}Mem and other representative methods. Subsequently, we compare token consumption between offline preprocessing and test-time inference, and investigate the test-time scaling of T^{2}Mem under different test-time token budgets. Finally, case studies are provided in Appendix A.6.

5.1 Overall Performance

In this subsection, we evaluate the overall performance of T^{2}Mem against a range of representative baselines. The results on LoCoMo are presented in Table 1, with additional evaluation on LongMemEval reported in Table 2. Overall, T^{2}Mem outperforms all baselines across both benchmarks, demonstrating the effectiveness of our approach to dynamically constructing memory at test time. Notably, T^{2}Mem delivers consistent gains across different backbone models, indicating its robustness and generalizability across model architectures.

Compared with retrieval-based methods such as Naive RAG and Search-o1, T^{2}Mem achieves higher average performance on both benchmarks. On LoCoMo, the gains are particularly evident on Multi-Hop and Temporal questions, suggesting that dynamically constructing memory according to evolving information needs is more effective than relying on a fixed set of retrieved chunks or iterative retrieval alone. T^{2}Mem also exhibits more consistent performance across task categories than offline memory construction methods. These methods compress and organize the interaction history into persistent memory representations to enable efficient access in future interactions, but the pre-constructed memory may not align equally well with different information needs. For example, on LongMemEval (Table 2), LightMem shows strong performance on several task categories but

Table 3: Ablation Study on the LoCoMo Dataset.

Method	# Token ($\times 10^3$)	Multi-Hop	Temporal	Open-Domain	Single-Hop	Avg.
Qwen3.5-27B						
T^2Mem (Summary)	6,809	57.80	71.96	<u>51.04</u>	73.01	68.64
T^2Mem (Graph)	7,943	52.13	59.19	45.83	69.92	62.92
T^2Mem	8,765	<u>59.22</u>	<u>75.70</u>	50.00	**76.81**	**71.69**
w/o Metadata	11,792	**59.92**	68.84	47.91	<u>75.86</u>	<u>69.74</u>
w/o Rawdata	7,415	56.03	**77.88**	**56.52**	68.85	67.60
w/o Rollout Construction	5,177	55.67	71.34	42.71	69.08	65.45
Gemma4-31B-Instruct						
T^2Mem (Summary)	5,245	<u>54.26</u>	**79.13**	48.96	75.74	<u>70.84</u>
T^2Mem (Graph)	7,133	52.48	67.91	43.75	70.04	64.74
T^2Mem	7,274	**56.74**	<u>77.26</u>	**52.08**	<u>76.10</u>	**71.30**
w/o Metadata	11,831	52.48	72.90	<u>50.00</u>	**78.48**	<u>70.84</u>
w/o Rawdata	6,668	<u>54.26</u>	**79.13**	<u>50.00</u>	66.94	66.10
w/o Rollout Construction	3,641	52.84	75.39	42.71	63.85	62.92

substantially lower accuracy on Single-Assistant questions. This suggests that such offline memory construction can lead to substantial performance variation across different task categories.

Furthermore, T^2Mem also outperforms LazyMem, which constructs memory from a one-pass retrieval set of query-relevant facts, making subsequent construction dependent on the quality and coverage of the initial retrieval. In contrast, T^2Mem unfolds memory construction along the reasoning process, progressively retrieving the evidence needed to construct and refine the memory. This enables T^2Mem to adaptively acquire missing evidence as reasoning unfolds, resulting in more comprehensive memory construction. Moreover, by supplying the necessary evidence at the appropriate stages of reasoning, T^2Mem avoids redundant reasoning and achieves stronger performance.

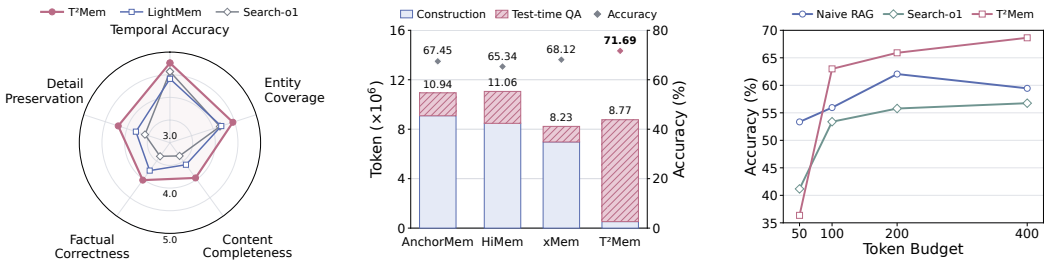
5.2 Ablation Studies

This subsection conducts ablation studies on LoCoMo to examine the contributions of different components in T^2Mem. Additional results on LongMemEval are reported in Appendix A.4.

Specifically, we first examine the effectiveness of the lightweight episode-level metadata proposed in Sec. 3.1 by comparing it with two alternative representations. T^2Mem (Summary) replaces the metadata for each session with a concise summary (Tan et al., 2025), while T^2Mem (Graph) represents the relations among entities and events within each session using a graph structure (Wu et al., 2026). We further investigate the roles of metadata and raw interactions during memory construction. T^2Mem (w/o Metadata) directly performs dynamic memory construction over the raw interaction sessions without using metadata, whereas T^2Mem (w/o Rawdata) relies solely on the metadata. Finally, T^2Mem (w/o Rollout Construction) removes the rollout memory construction process and directly answers the user query using the initial retrieval results. As shown in Table 3, T^2Mem consistently outperforms both T^2Mem (Summary) and T^2Mem (Graph), demonstrating the effectiveness of its semi-structured metadata. This advantage can be attributed to the complementary structural and semantic cues, such as timestamps, event information, and topics, which help LLMs capture dependencies among events during reasoning and identify relevant sessions for memory construction. Furthermore, the performance degradation in T^2Mem (w/o Metadata) and T^2Mem (w/o Rawdata) highlights the complementary roles of these two knowledge sources. The metadata provides cues for locating relevant interactions, while the raw sessions preserve fine-grained events for recovering detailed evidence. Removing the rollout memory construction process reduces token consumption, but leads to a clear performance degradation in T^2Mem (w/o Rollout Construction). This suggests that progressively integrating retrieved knowledge according to evolving information needs enables more effective and tailored memory construction.

5.3 Effectiveness of Test-Time Computation for Memory Construction

In this subsection, we analyze T^2Mem in terms of memory quality, token allocation, and test-time scaling, as illustrated in Figure 2. All experiments are conducted using Qwen3.5-27B on the LoCoMo dataset. Additional experimental details are provided in Appendix A.3.



(a) Quality of the Constructed Memory.

(b) Token Consumption for Memory Construction and Inference.

(c) QA Accuracy across Different Token Budgets.

Figure 2: Analysis of T^{2}Mem. We examine (a) the quality of the constructed memory, (b) token consumption during offline memory construction and LLM inference, and (c) QA performance across different test-time token budgets.

Quality of Constructed Memory. As shown in Figure 2a, we evaluate the quality of the memory constructed by T^{2}Mem, LightMem, and Search-o1 along five dimensions, with the evaluation criteria detailed in Appendix A.8. Specifically, *Temporal Accuracy* and *Entity Coverage* assess the ability to capture temporal dependencies and entity associations across interactions. *Content Completeness* and *Factual Correctness* measure whether the constructed memory contains sufficient information to answer the query and remains consistent with the ground-truth answer, respectively. Finally, *Detail Preservation* evaluates how well the memory retains specific quantities and other details. Across all five dimensions, T^{2}Mem achieves the highest scores, indicating that it constructs more complete and reliable memory. This advantage can be attributed to its ability to progressively capture episode dependencies, thereby helping retrieve more accurate evidence to facilitate the LLM in answering the query. In particular, the improvement in *Detail Preservation* suggests that T^{2}Mem retains finer-grained evidence beyond the core information needed to answer the query. This benefit stems from its iterative construction process, which allows the model to read additional metadata or access linked source sessions when the accumulated memory is insufficient to support construction. The higher *Factual Correctness* further demonstrates that T^{2}Mem can effectively incorporate supporting evidence, leading to better consistency with the ground-truth answers.

Association between Token Consumption and QA Accuracy. As shown in Figure 2b, T^{2}Mem achieves the highest QA accuracy while consuming a comparable number of tokens to the three offline memory construction systems. This efficiency stems from a different computation allocation strategy: rather than constructing comprehensive memories offline, T^{2}Mem extracts only lightweight metadata in advance and shifts the remaining computation to test time, allowing memory construction to adapt to the information needs of each query. To further investigate this property and its scaling behavior, we vary the token budget available to T^{2}Mem and compare it with Naive RAG and Search-o1. As shown in Figure 2c, T^{2}Mem consistently improves with additional test-time computation, whereas Search-o1 exhibits a weaker scaling trend and Naive RAG shows no consistent benefit, with its performance even declining as the token budget increases. These results indicate that effective test-time scaling depends not merely on increasing the computation budget, but on effectively translating additional computation into useful evidence for memory evolution.

6 Conclusion

This paper presents T^{2}Mem, a test-time memory construction framework for long-term conversational question answering. T^{2}Mem extracts lightweight episode-level metadata offline as retrieval cues while retaining links to the original interactions, and adaptively retrieves and organizes metadata and raw interactions at test time to progressively construct query-specific memory. Experiments on LoCoMo and LongMemEval show that T^{2}Mem consistently outperforms strong baselines across two backbone LLMs while substantially reducing memory construction overhead. Further analysis demonstrates that deferring memory construction to inference enables the memory construction process to dynamically adapt to the information needs of individual queries.

AI USE STATEMENT

Generative AI tools were used only for proofreading, including the correction of typographical and grammatical errors. All methodological designs, experimental execution, interpretation of the results, manuscript content, references, and citations were prepared and verified by the authors. The authors take full responsibility for the final content of this work.

REFERENCES

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of ACL*, pp. 2318–2335, 2024. URL https://aclanthology.org/2024.findings-acl.137/.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *ArXiv preprint*, abs/2504.19413, 2025. URL https://arxiv.org/abs/2504.19413.

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

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. LightMem: Lightweight and efficient memory-augmented generation. In *Proceedings of ICLR*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a05b72653ec5b473732129829ae04195-Abstract-Conference.html.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633–638, 2025. URL https://www.nature.com/articles/s41586-025-09422-z.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware llm reasoning. In *Findings of ACL*, pp. 24842–24855, 2025. URL https://aclanthology.org/2025.findings-acl.1274/.

Aidan J Horner, James A Bisby, Daniel Bush, Wen-Jing Lin, and Neil Burgess. Evidence for holistic episodic recollection via hippocampal pattern completion. *Nature communications*, pp. 7462, 2015. URL https://www.nature.com/articles/ncomms8462.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *Proceedings of COLM*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Chuanrui Hu, Xingze Gao, Zuyi Zhou, Dannong Xu, Yi Bai, Xintong Li, Hui Zhang, Tong Li, Chong Zhang, Lidong Bing, and Yafeng Deng. EverMemOS: A self-organizing memory operating system for structured long-horizon reasoning. In *Proceedings of ACL*, pp. 45836–45853, 2026a. URL https://aclanthology.org/2026.acl-long.2125/.

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, Zhenrong Cheng, Xuanbo Fan, Jiaxin Guo, Xinlei Yu, Zhenhong Zhou, Zewen Hu, Jiahao Huo, Junhao Wang, Yuwei Niu, Yu Wang, Zhenfei Yin, Xiaobin Hu, Yue Liao, Qiankun Li, Kun Wang, Wangchunshu Zhou, Yixin Liu, Dawei Cheng, Qi Zhang, Tao Gui, Shirui Pan, Yan Zhang, Philip Torr, Zhicheng Dou, Ji-Rong Wen, Xuanjing Huang, Yu-Gang Jiang, and Shuicheng Yan. Memory in the age of ai agents. *ArXiv preprint*, abs/2512.13564, 2025. URL https://arxiv.org/abs/2512.13564.

Zhanghao Hu, Qinglin Zhu, Runcong Zhao, Di Liang, Hanqi Yan, Yulan He, and Lin Gui. Beyond rag for agent memory: Retrieval by decoupling and aggregation. *ArXiv preprint*, abs/2602.02007, 2026b. URL https://arxiv.org/abs/2602.02007.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In *Proceedings of EMNLP*, pp. 25961–25970, 2025. URL https://aclanthology.org/2025.emnlp-main.1318/.

Ayeong Lee, Ethan Che, and Tianyi Peng. How well do llms compress their own chain-of-thought? a token complexity approach. *ArXiv preprint*, abs/2503.01141, 2025. URL https://arxiv.org/abs/2503.01141.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of NeurIPS*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Kai Li, Xuanqing Yu, Ziyi Ni, Yi Zeng, Yao Xu, Zheqing Zhang, Xin Li, Jitao Sang, Xiaogang Duan, Xuelei Wang, Chengbao Liu, and Jie Tan. TiMem: Temporal-hierarchical memory consolidation for long-horizon conversational agents. In *Findings of ACL*, pp. 21700–21720, 2026a. URL https://aclanthology.org/2026.findings-acl.1091/.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of EMNLP*, pp. 5420–5438, 2025. URL https://aclanthology.org/2025.emnlp-main.276/.

Xinze Li, Yuqing Lan, Zhenghao Liu, Haidong Xin, Yukun Yan, Shuo Wang, Zheni Zeng, Sen Mei, Ge Yu, and Maosong Sun. Seek: Steering llm reasoning for rag via internal reasoning sketches. *ArXiv preprint*, abs/2601.09402, 2026b. URL https://arxiv.org/abs/2601.09402.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *TACL*, pp. 157–173, 2024. URL https://aclanthology.org/2024.tacl-1.9/.

Zhenghao Liu, Pengcheng Huang, Zhipeng Xu, Xinze Li, Shuliang Liu, Chunyi Peng, Haidong Xin, Yukun Yan, Shuo Wang, Xu Han, Zhiyuan Liu, Maosong Sun, Yu Gu, and Ge Yu. Knowledge intensive agents. *AI Open*, pp. 18–44, 2026. URL https://www.sciencedirect.com/science/article/pii/S2666651026000033.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of ACL*, pp. 13851–13870, 2024. URL https://aclanthology.org/2024.acl-long.747/.

OpenAI, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Bohan Zhang, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wenting Zhan, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card. *ArXiv preprint*, abs/2412.16720, 2024. URL https://arxiv.org/abs/2412.16720.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *ArXiv preprint*, abs/2310.08560, 2023. URL https://arxiv.org/abs/2310.08560.

Chunyi Peng, Haidong Xin, Xuanshuo Sheng, Xin Dai, Zhenghao Liu, Shuo Wang, Yukun Yan, Zulong Chen, Yu Gu, and Ge Yu. Memory shot for long-term dialogue. *ArXiv preprint*, abs/2606.28338, 2026. URL https://arxiv.org/abs/2606.28338.

Qwen Team. Qwen3.5: Towards native multimodal agents, 2026. URL https://qwen.ai/blog?id=qwen3.5.

Zhanyu Shen, Sijie Cheng, Zhicheng Guo, Weiqin Wang, Yile Wang, and Hui Huang. AnchorMem: Anchored facts with associative contexts for building memory in large language models. In *Findings of ACL*, pp. 34784–34798, 2026. URL https://aclanthology.org/2026.findings-acl.1736/.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *Proceedings of ICLR*, pp. 10131–10165, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/1b623663fd9b874366f3ce019fdfdd44-Paper-Conference.pdf.

Jingyu Sun, Yuyang Xue, Mingyang Li, Zhengtao Yao, Jiachen Li, Yang Cui, Wenhao Cai, Haozhe Liu, Fangying Wang, Magdalene Katharina Montgomery, et al. Trajwiki: Source-grounded memory trajectories for long-horizon dialogue agents. *ArXiv preprint*, abs/2608.00967, 2026. URL https://arxiv.org/abs/2608.00967.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Rajan Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of ACL*, pp. 8416–8439, 2025. URL https://aclanthology.org/2025.acl-long.413/.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, Aditya Chawla, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku Evci, Clément Farabet, Johan Ferret, Filippo Galgani, Sertan Girgin, Jean-Bastien Grill, Maarten Grootendorst, Jiaxian Guo, Cassidy Hardin, Yanzhang He, Steven M. Hernandez, Omri Homburger, Léonard Hussenot, Juyeong Ji, Armand Joulin, Aishwarya Kamath, Parnian Kassraie, Olivier Lacombe, Preethi Lahoti, Gaël Liu, Gus Martins, Luciano Martins, Tatiana Matejovicova, Ramona Merhej, Nikola Momchev, Sneha Mondal, Ryan Mullins, Sindhu Raghuram Panyam, Shreya Pathak, Sarah Perrin, André Susano Pinto, Etienne Pot, Angéline Pouget, Alexandre Ramé, Sabela Ramos, Douglas Reid, David Rim, Morgane Rivière, Karsten Roth, Louis Rouillard, Omar Sanseviero, Pier Giuseppe Sessa, Shane Settle, Danila Sinopalnikov, Sara Smoot, Piotr Stanczyk, Andreas Steiner, Lawrence Stewart, Ilya Tolstikhin, Michael Tschannen, Anton Tsitsulin, Nino Vieillard, Renjie Wu, Pingmei Xu, Haichuan Yang, Edouard Yvinec, Biao Zhang, Li Zhang, Joe Zou, Nicolas Aagnes, Abdelrahman Abdelhamed, Jakub Adamek, Shivani Agrawal, Shubham Agrawal, Ibrahim Alabdulmohsin, Jean Baptiste Alayrac, Uri Alon, Chandramouli Amarnath, Ankesh Anand, Chrysovalantis Anastasiou, Setareh Ariafar, François-Xavier Aubet, Kyriakos Axiotis, Federico Barbero, Joelle Barral, Alexei Bendebury, Urs Bergmann, Stanley Bileschi, Kat Black, Mathieu Blondel, Sebastian Borgeaud, Arthur Bražinskas, Ryan Burnell, Robert Busa-Fekete, Mu Cai, Daniele Calandriello, Glenn Cameron, Charlotte Caucheteux, Rahma Chaabouni, Garima Chadha, Jetha Chan, Blake Jianhang Chen, Jesse Chen, Lin Chen, Xu Chen, Derek Cheng, Tzu hsiang Chien, Nikolai Chinaev, Yi Chou, Zhaohui Chu, Benjamin Coleman, Pooja Consul, Sam Conway-Rahman, Scott Crowell, Dylan Cutler, Vivek Dani, Samira Daruki, Anil Das, Daniel Deutsch, Nishanth Dikkala, Li Ding, Qiuhan Ding, Shenil Dodhia, Konstantin Donhauser, Tulsee Doshi, Anca Dragan, Alex Druinsky, Sahil Dua, Zoltan Egyed, Danielle Eisenbud, Daniel Eppens, Cindy Fan, Bahare Fatemi, Yassir Fathullah, Vlad Feinberg, Milen Ferev, Sebastian Flennerhag, Takumi Fujimoto, João Gabriel Oliveira, Isaac Galatzer-Levy, João Gante, Simon Geisler, Soham Ghosal, Antonious M. Girgis, Tamara von Glehn, Alec Go, Alhaad Gokhale, Alex Grills, Yiming Gu, Mayank Gupta, Pramod Gupta, Guru Guruganesh, Raia Hadsell, Hamza Harkous, Jitendra Harlalka, Demis Hassabis, Anja Hauth, Joe Heyward, Arian Hosseini, Chih-Yang Hsia, I-Hung Hsu, Xiaopeng Huang, Yangsibo Huang, Kevin Hui, Adrian Hutter, Te I, Fotis Iliopoulos, Advait Jain, Ganesh Jawahar, Ziwei Ji, Qilin Jin, Melvin Johnson, Kandarp Joshi, Arun Kandoor, Wang-Cheng Kang, Koray Kavukcuoglu, Mehran Kazemi, Kathleen Kenealy, Amr Khalifa, Phoebe Kirk, Ivan Korotkov, Suraj Kothawade, Vitaly Kovalev, Neel Kovelamudi, Adam Kraft, Ravin Kumar, Vivek Kumar, Harish Kuppam, Justin Lannin, Chen-Yu Lee, Seungji Lee, Dmitry Lepikhin, Alon Levkovitch, Dongdong Li, Qiujia Li, Valentin Liévin,

Ethan Lin, Ziqian Lin, Casper Liu, Tianlin Liu, Tianqi Liu, Xin Liu, Ivan Lobov, Mayank Lunayach, Min Ma, Gagan Madan, Andrii Maksai, Eric Malmi, Michal Matuszak, Daniel McDuff, Gaurav Menghani, Maciej Mikuła, Daniil Mirylenka, Karolis Misiunas, Vedant Misra, Andreea Mitran, Kareem Mohamed, Maksim Mukha, Eric Noland, James O'Donnell, Brendan O'Donoghue, Kate Olszewska, Bernett Orlando, Wanqiong Pan, Rina Panigrahy, Unnati Parekh, Nicolas Perez-Nieves, Chunjong Park, Eric Paskie, Liqian Peng, Bryce Petrini, Slav Petrov, Jonas Pfeiffer, Bilal Piot, Martyna Plomecka, Siim Poder, Octavio Ponce, Arijit Pramanik, David Racz, Anish Rajan, Michelle Ramanovich, Anand Rao, Marvin Ritter, Vitor Rodrigues, Evan Rosen, Mikołaj Rybiński, Noveen Sachdeva, Michaël E. Sander, Rohit Sathyanarayana, Sagar Savla, Samuel Schmidgall, Tal Schuster, George Scrivener, Benoit Seguin, Andrew Sellergren, Aliaksei Severyn, Izhak Shafran, Dhruv Shah, Bobak Shahriari, Yuan Shangguan, Ashish Shenoy, Pradeep Shenoy, Rakesh Shivanna, Pauline Sho, Lucas Spangher, Wojciech Stokowiec, Tim Strother, Yao Su, Yinghao Sun, Mukund Sundararajan, Andrea Tacchetti, Mor Hazan Taege, Pouya Tafti, Jean Tarbouriech, Chetan Tekur, Shantanu Thakoor, Rahul Thapa, Madeleine Traverse, Lenart Treven, Tao Tu, Chien Te Tung, Çağlar Ünlü, Petar Veličković, Malini Pooni Venkat, Sagar Gubbi Venkatesh, Vidya Venkiteswaran, Francesco Visin, Alex Vitvitskyi, Kiran Vodrahalli, Weiyi Wang, Xin Wang, Tris Warkentin, Jan Wassenberg, John Wieting, Cindy Wu, Lechao Xiao, Hao Xu, Yuhui Xu, Fuzhao Xue, Arun Yadav, Jun Yan, Antoine Yang, Lin Yang, Ming-Hsuan Yang, Ziyu Ying, Jae Hyeon Yoo, Morteza Zadimoghaddam, Sajjad Zafar, Fred Zhang, Jiageng Zhang, Jianyi Zhang, Xiaofan Zhang, Chao Zhao, David Zhou, and Chen Zou. Gemma 4 technical report. *ArXiv preprint*, abs/2607.02770, 2026. URL https://arxiv.org/abs/2607.02770.

Timothy J Teyler and Jerry W Rudy. The hippocampal indexing theory and episodic memory: updating the index. *Hippocampus*, pp. 1158–1169, 2007. URL https://pubmed.ncbi.nlm.nih.gov/17696170/.

Yanting Wang, Runpeng Geng, Ying Chen, and Jinyuan Jia. Attntrace: Contextual attribution of prompt injection and knowledge corruption. In *2026 IEEE Symposium on Security and Privacy (SP)*, pp. 4320–4338. IEEE, 2026.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *Proceedings of ICLR*, pp. 86809–86836, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/d813d324dbf0598bbdc9c8e79740ed01-Paper-Conference.pdf.

Mingyan Wu, Zhenghao Liu, Xinze Li, Yuqing Lan, Yukun Yan, Shuo Wang, Cheng Yang, Minghe Yu, Zheni Zeng, and Maosong Sun. Finding what matters: Anchoring context knowledge with evolving indices for iterative retrieval. *ArXiv preprint*, abs/2601.16462, 2026. URL https://arxiv.org/abs/2601.16462.

Haidong Xin, Xinze Li, Zhenghao Liu, Yukun Yan, Shuo Wang, Cheng Yang, Yu Gu, Ge Yu, and Maosong Sun. MetaMem: Evolving meta-memory for knowledge utilization through self-reflective symbolic optimization. In *Findings of ACL*, pp. 5473–5492, 2026a. URL https://aclanthology.org/2026.findings-acl.270/.

Haidong Xin, Zhenghao Liu, Sen Mei, Yukun Yan, Shi Yu, Shuo Wang, Zulong Chen, Yu Gu, Ge Yu, and Chenyan Xiong. Lisrec: Modeling user preferences with learned item shortcuts for sequential recommendation. In *Proceedings of KDD*, pp. 1650–1661, 2026b. URL https://doi.org/10.1145/3770854.3780337.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *Proceedings of ICLR*, 2023. URL https://openreview.net/pdf?id=WE_vluYUL-X.

Jianing Yin and Tan Tang. Defermem: Query-time evidence distillation via reinforcement learning for long-term memory qa. *ArXiv preprint*, abs/2605.22411, 2026. URL https://arxiv.org/abs/2605.22411.

Jing Yu, Yibo Zhao, Jiaming Zhang, and Xiang Li. Lazymem: Retrieve broadly, construct selectively for efficient long-term agent memory. *ArXiv preprint*, abs/2607.22690, 2026. URL https://arxiv.org/abs/2607.22690.

Ningning Zhang, Xingxing Yang, Zhizhong Tan, Weiping Deng, and Wenyong Wang. Himem: Hierarchical long-term memory for llm long-horizon agents. *ArXiv preprint*, abs/2601.06377, 2026. URL https://arxiv.org/abs/2601.06377.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model-based agents. *TOIS*, 43(6):1–47, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of NeurIPS*, volume 36, pp. 46595–46623, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of AAAI*, pp. 19724–19731, 2024. URL https://doi.org/10.1609/aaai.v38i17.29946.

A APPENDIX

A.1 LICENSE

All datasets used in this work are publicly available with licenses permitting academic research and non-commercial use. Specifically, LoCoMo is distributed under the Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0), while LongMemEval is released under the MIT License.

Table 4: Statistics of the LoCoMo and the LongMemEval Datasets.

LoCoMo		**LongMemEval**	
Category	**# Questions**	**Category**	**# Questions**
Single-Hop	841	Single-Session User	70
Multi-Hop	282	Single-Session Assistant	56
Temporal	321	Single-Session Preference	30
Open Domain	96	Multi-Session	133
-	-	Knowledge-Update	78
-	-	Temporal-Reasoning	133
Total	1,540	Total	500

A.2 ADDITIONAL DATASET DETAILS

Table 4 summarizes the statistics of the two long-term memory benchmarks adopted in our experiments: LoCoMo (Maharana et al., 2024) and LongMemEval (Wu et al., 2025). Following prior studies (Fang et al., 2026; Hu et al., 2026a), we exclude the `adversarial` category from LoCoMo for evaluation, and use the official `longmemeval_s_cleaned` split[1] for LongMemEval.

A.3 ADDITIONAL EXPERIMENTAL DETAILS

In all experiments, we employ vLLM[2] to serve both the backbone and retrieval models. For generation, we set the temperature to 0.0, top-p to 0.95, and top-k to 20, with a maximum answer length of 1024 tokens. For HiMem (Zhang et al., 2026), we follow the original implementation and adopt bge-reranker-v2-m3 (Chen et al., 2024) as the reranking model. For LazyMem (Yu et al., 2026), we replace the original memory-processing model, LazyMem-4B, with the corresponding backbone model, since LazyMem-4B had not been publicly released when our experiments were conducted. The reported LazyMem results therefore correspond to this backbone-substituted implementation.

For the token-budget analysis, we adopt prompt-based length control following prior work (Han et al., 2025; Lee et al., 2025). Specifically, the model is instructed to generate at most N reasoning tokens before producing the final answer. We override the default output limit by setting `max_tokens` to $N + 64$, where the additional 64 tokens provide a fixed allowance for the final answer and output formatting. This limit applies to the complete generated response and does not separately enforce the reasoning length. The final-answer instructions remain unchanged across all budget settings. Native thinking mode is disabled, and reasoning is generated explicitly in the response.

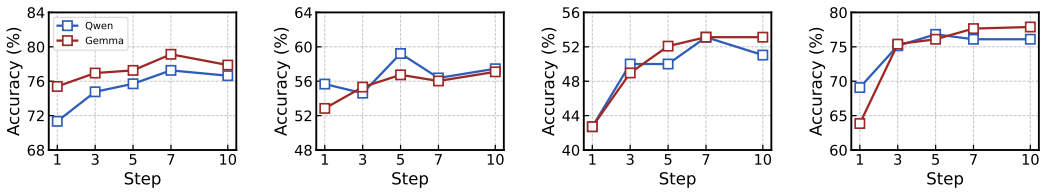
A.4 ABLATION STUDIES ON THE LONGMEMEVAL DATASET

As shown in Table 5, T^{2}Mem achieves the highest average accuracy on LongMemEval with both backbone models. Replacing episode-level metadata with summaries or graphs reduces average performance, supporting the effectiveness of the proposed representation. Removing either metadata or access to raw interactions also lowers average accuracy, highlighting their complementary roles in memory construction. Removing rollout construction leads to clear declines on Temporal Reasoning

[1] `https://huggingface.co/datasets/xiaowu0162/longmemeval`
[2] `https://vllm.ai/`

Table 5: Ablation Study on the LongMemEval Dataset.

Method	Single User	Single Assistant	Single Preference	Multi Session	Knowledge Update	Temporal Reasoning	Avg.
Qwen3.5-27B							
T^{2}Mem (Summary)	85.71	94.64	60.00	60.90	82.05	73.68	74.80
T^{2}Mem (Graph)	84.29	**100.00**	70.00	59.40	82.05	79.70	77.00
T^{2}Mem	92.86	98.21	**83.33**	70.68	92.31	**84.96**	**84.80**
w/o Metadata	90.00	94.64	53.33	62.41	89.74	75.19	77.00
w/o Rawdata	91.43	98.21	76.67	**72.93**	89.74	79.70	83.00
w/o Rollout Construction	**94.29**	94.64	73.33	69.17	**93.59**	79.70	82.40
Gemma4-31B-Instruct							
T^{2}Mem (Summary)	91.43	**100.00**	76.67	62.41	80.77	70.68	76.60
T^{2}Mem (Graph)	91.43	**100.00**	76.67	70.68	91.03	81.95	83.40
T^{2}Mem	94.29	**100.00**	**83.33**	69.92	**92.31**	**85.71**	**85.20**
w/o Metadata	94.29	94.64	56.67	**73.68**	89.74	84.96	84.00
w/o Rawdata	**95.71**	94.64	76.67	69.17	91.03	83.46	83.40
w/o Rollout Construction	94.29	91.07	73.33	66.92	91.03	80.45	81.20



(a) Temporal (b) Multi-Hop (c) Open-Domain (d) Single-Hop

Figure 3: Performance of T^{2}Mem at Different Memory Construction Steps on the LoCoMo Dataset.

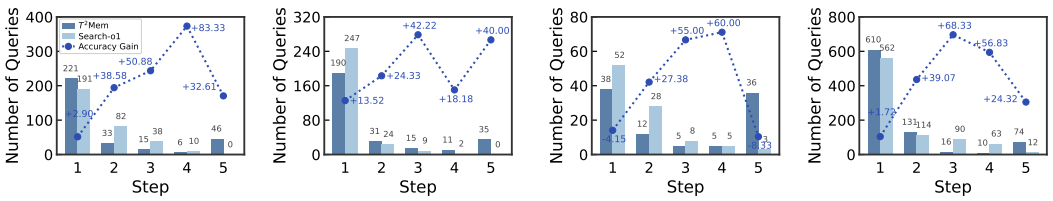
and Single Preference questions. These results further support progressively constructing memory according to the query's evolving information needs.

A.5 Effectiveness of T^{2}Mem at Different Memory Construction Steps

We conduct two analyses on LoCoMo to examine the effect of memory construction steps. We first vary the maximum construction budget $T_{\max}$ to evaluate its effect on answer accuracy. We then examine the actual number of steps T and compare T^{2}Mem with Search-o1 across observed stopping steps. For T^{2}Mem, each construction step generates (r_t, m_t, a_t) and appends the new memory cache m_t to the accumulated memory.

Effect of Construction Steps on Answer Performance. As shown in Figure 3, the answer accuracy generally improves as the maximum number of construction steps increases, indicating that additional test-time computation benefits memory construction. Specifically, Single-Hop questions show clear gains when the number of steps increases from one to three, followed by a clear plateau, as their required evidence can often be recovered within relatively shallow construction. Temporal and Open-Domain questions exhibit a different trend, with performance continuing to improve as the construction depth increases to seven steps. These questions typically require resolving temporal relations or recovering evidence distributed across interaction sessions, leaving more room for additional construction steps to contribute. Beyond this depth, the gains become less pronounced, reflecting diminishing returns from further extending the construction process.

Performance Gains of T^{2}Mem over Search-o1 at Different Construction Steps. Figure 4 further compares T^{2}Mem and Search-o1 according to the number of construction steps taken before answering. At the first step, the performance gap remains relatively small, particularly for Single-Hop questions, where the initially retrieved evidence is often sufficient to answer. However, as the process extends to two and three steps, T^{2}Mem achieves substantially larger gains across all types of questions. Although Search-o1 performs iterative retrieval, its subsequent searches are directly driven



(a) Temporal (b) Multi-Hop (c) Open-Domain (d) Single-Hop

Figure 4: Performance Gains of T^{2}Mem over Search-o1 across Different Memory Construction Steps on the LoCoMo Dataset.

by the current reasoning state, without an explicit structure for organizing previously encountered events, entities, and temporal cues. In contrast, T^{2}Mem uses the lightweight episode-level metadata to provide structured cues for locating relevant sessions and selectively recovers fine-grained details from the linked source interactions as construction proceeds. This structured access allows successive steps to more effectively connect newly acquired evidence with information established earlier, which becomes increasingly beneficial when answering requires evidence scattered across multiple sessions.

A.6 Case Studies

In Figure 5, we present a qualitative case study to illustrate how T^{2}Mem performs iterative memory construction to resolve temporal ambiguity across interaction sessions.

The example involves a long-horizon conversational query where relevant evidence is scattered over multiple interaction sessions. The initial observation o_0 contains episode-level metadata entries that mention a planned camping trip in June and a later report of camping. These entries do not establish when the June trip actually occurred. T^{2}Mem therefore performs additional metadata retrieval and source-session recovery to seek more specific temporal evidence. A subsequently retrieved entry from the June 27 session states that Melanie went camping in the mountains during the preceding week. The model distills this temporal information into a memory cache and appends it to the accumulated memory. The resulting $\mathcal{M}_T$ supports the final answer, illustrating how iterative memory construction incorporates answer-relevant information that is missing from the initial observation.

A.7 Saliency Analysis of Answer-Supporting Evidence

As shown in Figure 6, we compare the displayed contexts, token saliency maps, and answers of T^{2}Mem and LightMem (Fang et al., 2026) on the same LoCoMo query.

Following the token-level attention aggregation described by Wang et al. (2026), we compute saliency through a teacher-forced forward pass over each input-output sequence. For an input token x_i, its saliency is:

$$S(x_i) = \frac{1}{|\mathcal{L}_{\text{attn}}|\,|\mathcal{H}|\,|\mathcal{O}|} \sum_{\ell \in \mathcal{L}_{\text{attn}}} \sum_{h \in \mathcal{H}} \sum_{j \in \mathcal{O}} A_{j,i}^{(\ell,h)}, \tag{14}$$

where $A_{j,i}^{(\ell,h)}$ denotes attention from output position j to input token x_i. We average over the 16 softmax-attention layers $\mathcal{L}_{\text{attn}}$, their 24 query heads $\mathcal{H}$, and all output positions $\mathcal{O}$. The analyzed output is the recorded answer for LightMem and the controller JSON reconstructed from logs for the displayed T^{2}Mem step. Scores are normalized separately for visualization and interpreted within each input.

In Figure 6a, LightMem retrieves a preconstructed description of Caroline's relocation. Although this statement receives relatively high saliency, it identifies her origin only as "her home country," and the answer repeats this unresolved reference. The description preserves the relevant event but does not identify the country requested by the query. Thus, a salient and relevant memory can still lack the specificity needed to support a complete answer.

In contrast, T^{2}Mem constructs memory according to the current query, selecting and accumulating relevant facts during reasoning and acquiring additional information when needed. Figure 6b

Question: ***When did Melanie go camping in June?*** ***Golden Answer:*** ***The week before 27 June 2023***

<operation>**search_meta** (conv-26_session_6, conv-26_session_9)</operation>
<reasoning>
In session 2 (May 25), Melanie mentioned she was considering a camping trip in June. However, in session 9 (July 17), Melanie went camping with her family "two weekends ago", which refers to around July 3, 2023.
I need to determine whether the July camping event was the same trip she planned for June, or whether she had another camping trip in June.
</reasoning>
<memory_update>
[25 May] Melanie is considering a camping trip in June 2023.
[17 Jul] Melanie went camping with her family two weekends prior to July 17, which corresponds to around July 3, 2023.
</memory_update>

<operation>**read_raw** (conv-26_session_6)</operation>
<reasoning>
The retrieved wiki mentions that Melanie spent time with her children and went camping at the beach.
To verify whether this camping event corresponds to the June trip, I need to inspect the original conversation and check whether an exact camping date is provided.
</reasoning>
<memory_update>
[6 Jul] Melanie spent time with her children at a museum and camping at the beach.
However, the transcript does not provide an exact camping date.
</memory_update>

<operation>**search_meta** (conv-26_session_4)</operation>
<reasoning>
I have evidence that Melanie was considering a camping trip in June, and evidence that she went camping around July 3.
However, I still do not have a specific date for a camping trip that actually occurred in June.
I need to search additional memories for another mention of the June trip.
</reasoning>
<memory_update>
[25 May] Considering camping in June.
[17 Jul] Camping around July 3.
The temporal conflict remains unresolved.
</memory_update>

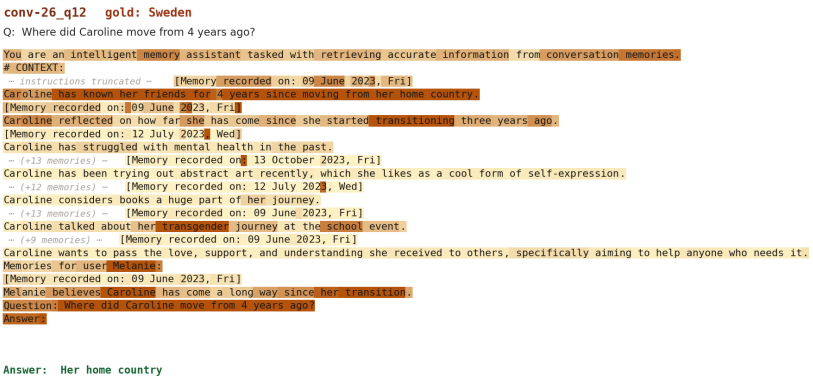
<operation>**answer**</operation>
<reasoning>
The retrieved wiki explicitly states that Melanie went camping in the mountains with her family during the week prior to June 27, 2023.
This provides a direct temporal anchor and resolves the previous ambiguity.
</reasoning>
<memory_update>
[27 Jun] Melanie went camping in the mountains the week prior to June 27, 2023.
Resolved date: June 19-25, 2023.
</memory_update>

Figure 5: Case Study of T^2Mem.

shows a salient metadata statement identifying Sweden as Caroline's home country. This explicit association supplies the relation needed to resolve the country reference, and the reported answer correctly names Sweden. The comparison illustrates the value of organizing memory around the query's specific evidence requirements rather than relying on a generally relevant event description alone.

A.8 Prompt Templates

This section presents the prompt templates used throughout our experiments. Figure 7 presents the prompt for generating an entry in the lightweight episode-level metadata from an interaction session.



(a) LightMem with Retrieved Memory Units.



(b) T^{2}Mem with Constructed Memory.

Figure 6: Token Saliency and Answer Specificity for LightMem and T^{2}Mem on the Same LoCoMo Query.

Figure 8 presents the prompt for producing the reasoning trace r_t, memory cache m_t, and next action a_t at each construction step. Figure 9 presents the prompt for generating the final answer from the query and the constructed memory $\mathcal{M}_T$. Figures 10 and 11 present the LLM-judge prompts for evaluating answer accuracy. Finally, Figure 12 presents the prompt for evaluating the quality of the answer-supporting evidence.

<system prompt>
You turn a raw conversation session into grounded wiki content for long-term conversational memory. Use only the provided transcript. Do not invent facts. Return ONLY valid JSON.
</system prompt>

<user prompt>
Create the content section for a grounded wiki page from this conversation session.

Authoritative session date/time:
{date_time}

Transcript:
{transcript}

1. Use the authoritative conversation date/time as the temporal anchor, resolving relative or ambiguous time expressions (e.g., yesterday, last year, next month) into explicit dates whenever possible.
2. Preserve concrete details that may be useful for later QA, including preferences, objects, locations, plans, family details, activities, and quoted names etc.
3. Write 8-15 atomic facts when the transcript contains enough information.

Return exactly this JSON schema:
{{
"summary": "2-3 sentence knowledge-style overview of the session's core event",
"event_core": ["core point 1", "core point 2", ...],
"atomic_facts": ["specific evidence-backed fact 1 ",
"specific evidence-backed fact 2", ...],
"entities": ["entity 1", "entity 2", ...],
"temporal_clues": ["temporal clue 1", ...],
"resolved_temporal_facts": ["event with resolved or anchored time", ...],
"ongoing_threads": ["thread 1", ...]
}}
</user prompt>

Figure 7: Prompt Template for Constructing Wiki-style Metadata from Interaction Sessions.

<system prompt>
You are an adaptive controller for long-term memory, choosing the next read action to maximize useful evidence.

Memory sources:
- wiki memories: wiki based on the raw session, highly condensed from the session topics and timeline clues.
- raw transcripts: compared to wikis, raw session content contains more precise evidence.

Allowed actions:
1. SEARCH_WIKI
2. READ_RAW_FROM_WIKI
3. ANSWER

Action guidance:
- Use `SEARCH_WIKI` when you need more wiki page for high-level abstract events, preferences, or timeline clues.
- Use `READ_RAW_FROM_WIKI` when provided wiki finds the right session but lacks exact detail, to inspect the raw session corresponding to that wiki entry.
- Terminate with `ANSWER` only when you already have enough support.
</system prompt>

<user prompt>
Decide the next reading action.

Question:
{question}

Current information:
{current_information}

Return exactly one JSON object with this schema:
```
{{
  "evidence": [
    {{
      "snippet": "Return copy the exact relevant span from the provided current information.",
      "date_time": "Return the exact date/time expression associated with the snippet, copied verbatim from the provided source text if available.",
    }},
    ...
  ],
  "reasoning": "Thinking about the current information and what action to choose",
  "action": "SEARCH_WIKI | READ_RAW_FROM_WIKI | ANSWER",
  "target_wiki_id": ["You need to view the original session's wiki ID.","xxxx",...],
}}
```

Notes: For `READ_RAW_FROM_WIKI`, fill `target_wiki_id` with the wiki ID whose raw session should be viewed.
</user prompt>

Figure 8: Prompt Template for Iterative Reasoning and Information Access During Test-time Memory Construction.

You are an intelligent assistant with access to a memory system. Your task is to answer the user's question by effectively utilizing the provided memory evidences.

Question:
{question}

Memory evidences:
{context}

 # Memory Evidences Guidelines:
1. Carefully analyze all provided memory evidences.
2. If the evidences contain contradictory information, prioritize the most recent dated evidence.
3. For relative time references in temporal questions, such as "yesterday", "two months ago", etc., resolve them against the memory Date and use clear formats such as "7 May 2023", "2022", or "The week before 6 July 2023".
4. The answer should usually be less than 5-8 words.

Figure 9: Prompt Template for Answering the User Question.

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".

Figure 10: Prompt Template for Response Judgement on the LoCoMo Dataset.

<TASK: SS-User, SS-Asst, Multi-S>
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.

<TASK: Temp. Reas>
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.

<TASK: Know. Upd>
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.

<TASK: SS-Pref>
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.

<TASK: Abstention>
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.

Figure 11: Prompt Template for Response Judgement on the LongMemEval Dataset.

You are an expert evaluator proficient in cognitive linguistics and natural language processing tasks. Your task is to evaluate the quality of a memory system's constructed memory, specifically its ability to support answering a given question correctly.

 # Input Data
*- **User Question**: **{user_question}***
*- **Ground Truth Answer**: **{ground_truth_answer}***
*- **Evaluated Evidence Context**: **{evaluated_evidence_context}***

 # Evaluation Criteria
*Please strictly review the "Evaluated Constructed Memory" based on the following five core dimensions. **You must assign a separate score (0-5) for each dimension.***

*1. **Temporal Accuracy***
- How accurately does the evidence capture dates, times, and temporal relationships?
- Are relative time expressions (e.g., "last year", "recently") resolved into explicit dates when possible?
*- *Negative Indicators*: Vague temporal references, missing dates, temporal contradictions.*

*2. **Entity Coverage***
- Are all relevant entities (people, places, objects, names, organizations) present in the evidence?
- Does the evidence capture the complete set of entities required to fully answer the question?
*- *Negative Indicators*: Missing key entities, partial entity coverage, wrong entities present.*

*3. **Content Completeness***
- Does the evidence contain sufficient information to derive the Ground Truth Answer?
- Are multiple relevant documents synthesized to cover all aspects of the question?
*- *Negative Indicators*: Information gaps, cannot infer answer from evidence, critical details missing.*

*4. **Factual Correctness***
- Are the facts in the evidence correct and aligned with the Ground Truth Answer?
- Does the evidence avoid including misleading or contradictory information?
*- *Negative Indicators*: Factual errors, contradiction with Ground Truth, hallucinated information.*

*5. **Detail Preservation***
- How faithfully does the evidence preserve specific details (exact wording, precise numbers, proper names)?
- Has the raw material been preserved at sufficient granularity, rather than over-summarized?
*- *Negative Indicators*: Over-generalization, loss of distinguishing details, paraphrase drift.*

 # Scoring Scale (Applied to EACH dimension independently)
*- **5 (Perfect)**: The dimension is executed flawlessly. Evidence is precise, complete, and directly supports the answer.*
*- **4 (Excellent)**: Strong execution. Meets high standards with only negligible gaps.*
*- **3 (Satisfactory)**: Acceptable. Contains main elements but lacks depth or precision.*
*- **2 (Poor)**: Weak. Significant gaps exist (e.g., missing key evidence, vague references, factual errors).*
*- **1 (Unusable)**: Severe failure. Context fails almost completely in this dimension.*
*- **0 (Failure)**: Completely irrelevant, empty, or non-existent effort in this dimension.*

 # Output Format
Return ONLY a JSON object with this exact structure:
```
{
  "temporal_accuracy": <int 0-5>,
  "entity_coverage": <int 0-5>,
  "content_completeness": <int 0-5>,
  "factual_correctness": <int 0-5>,
  "detail_preservation": <int 0-5>,
  "brief_justification": "<1-2 sentences summarizing the most salient finding>"
}
```

Figure 12: Prompt Template for Evaluating the Quality of Constructed Memory.