CreaMem: A Scene-Aware Memory Architecture for Personalized Agents

Qixuan Sun[1,3], Yue Que[1], Bowei He[2], Jin Guo[3,*], Dihang Yang[3], Wenchang Situ[3], Chen Ma[1,*]
[1]City University of Hong Kong, Hong Kong SAR, China
[2]Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
[3]AgentWoods Inc., San Francisco
{qixuansun2-c,yueque2-c}@my.cityu.edu.hk, Bowei.He@mbzuai.ac.ae
{jin,dihang,wenchang.situ}@creaseed.ai, chenma@cityu.edu.hk
[*]**Corresponding authors**

Abstract

Long-term memory is a core capability for personalized LLM agents. To support it, existing memory systems organize information using various criteria such as topic segments or summary hierarchies. However, we identify two major limitations in these designs. First, they lack scene awareness: memories from unrelated life scenes share the same retrieval space, which inflates the search space and introduces cross-scene interference. Second, they encode each memory from a single perspective, making it difficult to retrieve complementary views of the same event. In this paper, we propose the CreaMem architecture, which enables scene-aware memory organization by partitioning memory into several Life Scene Memories to reduce cross-scene interference at retrieval. To go beyond the single perspective and achieve cross-memory synergy, entries are dual-coded from both episodic and trait-based perspectives within each memory. We further devise a per-memory balanced sampling strategy at retrieval time. Extensive experiments on two long-term memory benchmarks show that CreaMem improves QA accuracy across all evaluation metrics, with particularly large gains on multi-hop reasoning performance, validating scene-aware partitioning and cross-memory synergy. To enhance reproducibility, we release our code in a public GitHub repository.

1 Introduction

Large language models (LLMs) are evolving from single-session tools into long-term personalized agents that accompany users across extended interactions (Li et al., 2024). A core capability of such agents is long-term memory, which enables them to retain, organize, and retrieve information from prior interactions, thereby maintaining coherence across sessions (Wang et al., 2024; Guo et al., 2024). However, retention alone is insufficient. As interactions accumulate over weeks and months, the memory structure determines how the information is organized and whether the right information can be retrieved. Consequently, the architectural design of memory organization and retrieval plays a significant role in sustaining coherent long-term behavior across extended interactions (Packer et al., 2023; Park et al., 2023), thus emerging as a central design problem for modern agent systems.

Existing memory systems address this design problem through various organization principles, which can be broadly categorized into three types: (1) Flat-storage methods (Song et al., 2020; Izacard et al., 2022; Lee et al., 2023) retrieve from an unstructured pool; (2) Structure-imposing methods (Pan et al., 2025; Sarthi et al., 2024; Wang et al., 2025; Gutiérrez et al., 2025; Xu et al., 2025; Xu et al., 2026) organize memory through explicit structures such as summary trees or entity graphs; and (3) Memory-partitioning methods (Li et al., 2026; Kang et al., 2025) decompose memory into several specialized components along axes like cognitive function or temporal scale. For example, MIRIX (Wang and Chen, 2025) separates memory into episodic, semantic, and procedural memories.

Although these methods differ in their organization principles, they overlook two aspects: none organizes memory by the user's life scenes, and few encode each experience along both an episodic and a trait-based perspective. These two gaps echo two well-established findings in cognitive psychology: (1) autobiographical memory is organized around lifetime contexts rather than abstract categories (Conway and Pleydell-Pearce, 2000; Conway, 2005); (2) episodic and semantic memory encode the same experience in complementary forms that cooperate during recall (Tulving, 1972, 2002).

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 17800–17813
October 24-29, 2026 ©2026 Association for Computational Linguistics

We adopt both as design principles: memory is partitioned along the user's life scenes, and each experience is encoded twice. It is stored as a timeline entry in the Episodic Memory and as a trait entry in the relevant scene memory. For example, when a user mentions a weekend hike with their family, the system stores an episodic entry such as "the user went hiking on May 7" with surrounding event details, alongside a Life Scene trait entry such as "the user spends weekends on family outdoor activities." Traits are kept within scenes because user preferences are context-dependent: the same kind of event in a work setting would yield a different trait, and pooling them together would wash out these scene-specific patterns. At retrieval, a temporal query such as "when did the user go hiking?" is answered by the Episodic Memory entry alone, while a broader query such as "is the user likely to enjoy a camping trip?" draws from both memories, returning the event and its scene-specific trait as mutually complementary perspectives.

In this paper, we instantiate these principles as CreaMem, a scene-aware memory architecture for long-term personal agents. At storage time, a Meta Memory Manager routes each incoming message to the relevant scene memory and adds a paired entry to the Episodic Memory. At retrieval time, queries are issued to all memories, and a per-memory balanced sampling strategy ensures that both the timeline view and the scene-specific view contribute to the final returned context for response generation.

In summary, our contributions are threefold.

- We partition memory along the user's life scenes, i.e., Life, Work, and Interest, together with a dedicated Episodic Memory, thereby reducing cross-scene interference at retrieval.
- We encode each experience as both an episodic entry in the Episodic Memory and a trait entry in the relevant scene memory, and combine them at retrieval through per-memory balanced sampling.
- Experiments on two long-term memory benchmarks show that CreaMem improves overall QA accuracy and multi-hop reasoning performance; ablation study further indicates that cross-memory synergy, not any single component, accounts for the majority of the observed gain.

2 Related Work

CreaMem draws from two bodies of prior work: agent memory systems for LLMs, and cognitive science theories of human memory organization.

2.1 Memory for LLM Agents

LLMs face fundamental challenges in handling complex scenarios that require long-term coherence, where fixed-length contexts struggle to maintain continuity across dialogues with temporal gaps (Wang et al., 2024; Guo et al., 2024; Wu et al., 2026). Full-context approaches (Chen et al., 2023; Brown et al., 2020; OpenAI, 2023; Touvron et al., 2023) place the dialogue history directly in the LLM's context window, but scale poorly as histories grow beyond the window (Gao et al., 2024; Liu et al., 2024; Paulsen, 2025).

External memory systems address this by extracting and retrieving relevant content on demand. Flat-storage methods such as MemGPT (Packer et al., 2023), MemoryBank (Zhong et al., 2024), MPNet (Song et al., 2020), Contriever (Izacard et al., 2022), and MPC (Lee et al., 2023) index memory entries in an undifferentiated pool for RAG-style retrieval (Lewis et al., 2020), where unrelated entries compete for retrieval slots. Structure-imposing methods mitigate this by adding summary hierarchies (Wang et al., 2025; Sarthi et al., 2024), topical segments (Pan et al., 2025), knowledge graphs (Gutiérrez et al., 2025; Rasmussen et al., 2025; Pan et al., 2024; Hamilton et al., 2017), or note-based links (Xu et al., 2025; Chhikara et al., 2025), yet remain confined to a single memory pool where unrelated contexts still compete.

Most closely related to our work, memory-partitioning architectures (Wang and Chen, 2025; Kang et al., 2025; Li et al., 2026; Feng et al., 2026; Hu et al., 2026; Lin et al., 2026; Zhou et al., 2025; Lei et al., 2026; Tiwari and Fofadiya, 2026; Yang et al., 2026; Yu et al., 2026) decompose memory into specialized components along axes such as temporal scale or cognitive function. For example, MemoryOS (Kang et al., 2025) organizes memory into short-, mid-, and long-term levels, while MIRIX (Wang and Chen, 2025) separates episodic, semantic, and procedural memory. This paradigm leaves two aspects unaddressed: memories from different life scenes still share the same space at retrieval, and each experience is encoded from a single perspective without a mechanism for combining complementary views.

2.2 Cognitive Foundations

Two complementary lines of cognitive psychology inform CreaMem's design. Conway's Self-Memory System (Conway and Pleydell-Pearce,

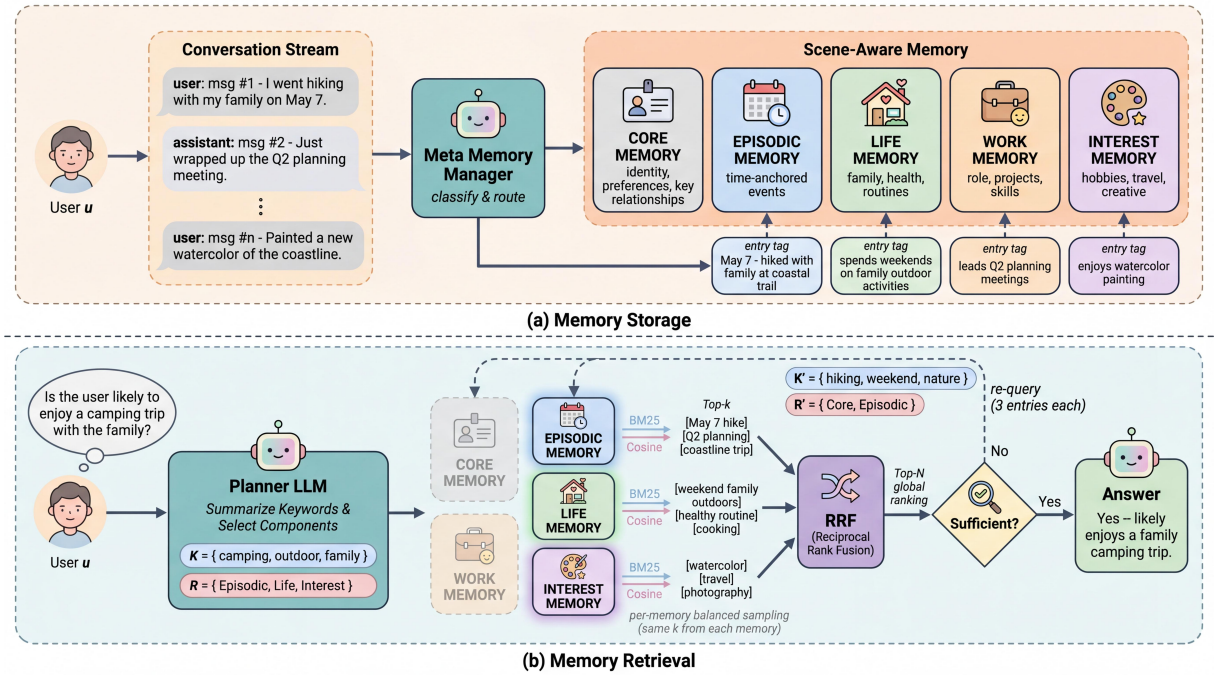

Figure 1: Overview of the CreaMem architecture, consisting of a Meta Memory Manager, an Episodic Memory, three Life Scene Memories, and a Core Memory.

2000; Conway, 2005) observes that autobiographical memories are organized by lifetime contexts rather than abstract categories, while Tulving's multiple memory systems (Tulving, 1972, 2002) further observes that episodic and semantic memory encode the same experience from complementary perspectives and cooperate during retrieval.

Inspired by these observations, CreaMem partitions memory along user life scenes and stores each experience as a timeline entry in the Episodic Memory and a trait entry in the relevant scene memory.

3 CreaMem

CreaMem is a scene-aware memory architecture for long-term personal agents that organizes user interactions along user's life scenes and combines complementary timeline with trait perspectives at retrieval, supporting coherent and personalized responses across extended conversations.

3.1 Overview Architecture

The overall architecture of CreaMem is illustrated in Figure 1. It consists of three modules: memory components, storage and retrieval.

Memory Components: This module defines the structural organization of memory in CreaMem. Memory comprises three components: an Episodic Memory that stores timeline entries of past events, a set of Life Scene Memories, Life, Work, and Interest, that store trait entries reflecting the user's behavior and preferences within each life scene, and a Core Memory that maintains a holistic user profile spanning all scenes.

Memory Storage: This module handles how each incoming message is written into memory. A Meta Memory Manager routes each message to the relevant Life Scene Memory and simultaneously adds a paired timeline entry to the Episodic Memory, recording each experience along both temporal and trait perspectives.

Memory Retrieval: This module retrieves relevant entries for a given query. Queries are issued to all selected memory components, and a per-memory balanced sampling strategy ensures that both the timeline view and the scene-specific view contribute to the returned context.

3.2 Memory Components

Meta Memory Manager. The Meta Memory Manager serves as the central router of CreaMem. Upon receiving an incoming message, it classifies the content into one or more relevant scenes and dispatches the segment to the corresponding memories. Because a single message often carries information across multiple life scenes, e.g. "I went to the gym with my colleague after work", the Manager may route the same message to Life, Work, and Interest Memories simultaneously.

Planner LLM. Given a user question, the Planner LLM selects memory components to query, outputs a set of keywords to drive retrieval, and generates the final response.

Core Memory. Core Memory maintains a stable user profile across scenes, storing identity attributes, preferences, key relationships, and other static reference information. It is injected into the prompt in full when selected by the Planner LLM.

Episodic Memory. Episodic Memory stores time-anchored events together with their surrounding narrative details. Each entry records a specific occurrence, such as a hike on 7 May, a promotion announcement, or concert attendance, along with the contextual details from which it was drawn. This component supports queries that require concrete grounding in particular past moments.

Life Scene Memories. The three Life Scene Memories capture trait-level knowledge about the user within a distinct life scene. **Life Memory** covers personal-life aspects such as family, health, daily routines, and food. **Work Memory** covers professional and academic aspects such as the user's role, ongoing projects, skills, and career goals. **Interest Memory** covers leisure aspects such as hobbies, entertainment, travel, and creative pursuits. Each entry within a Life Scene Memory is a trait: a distilled, scene-conditional inference about the user, abstracted from one or more underlying messages in the dialogue history.

3.3 Memory Storage

Each incoming message m is processed through a three-step pipeline. The Meta Memory Manager first classifies m into a subset of relevant memory components $R(m) \subseteq \{\text{Core}, \text{Episodic}, \text{Life}, \text{Work}, \text{Interest}\}$. For each selected memory component $r \in R(m)$, a corresponding LLM-based extractor f^{r}_{LLM} produces candidate memory entries. For each candidate, the system retrieves the most similar existing entries via embedding-based lookup, and the LLM determines whether to merge the candidate into an existing entry or insert it as a new entry into the corresponding SQLite table.

Routing policy. The Manager follows a *wide-entry, strict-filtering* policy: when scene classification is uncertain, multiple memory components are triggered to prevent silent information loss. A single message may therefore be routed to several memory components at once.

Core Memory. Core Memory stores the user's profile as a free-text block. For each routed message, the LLM appends an extracted update $v = f^{\text{core}}_{\text{LLM}}(m)$ to the block. When the block reaches 90% of its character limit, the LLM automatically compresses it to free space for further updates.

Episodic Memory. Each Episodic entry is a structured tuple

$$M^{(i)}_{\text{epi}} = \big(\,\text{eid}_i,\ t_i,\ a_i,\ e_i,\ s_i,\ d_i,\ \mathbf{v}_{s,i},\ \mathbf{v}_{d,i}\,\big), \tag{1}$$

where eid_i is a unique episodic entry identifier, t_i is the event timestamp, $a_i \in \{\text{user}, \text{assistant}\}$ the actor, e_i the event type, s_i a one-sentence summary, d_i the full event context, and $\mathbf{v}_{s,i} = f_{\text{emb}}(s_i)$, $\mathbf{v}_{d,i} = f_{\text{emb}}(d_i)$ are dense embeddings of the summary and details produced by an embedding function f_{emb}. A single message may yield multiple entries:

$$\{M^{(i)}_{\text{epi}}\}_{i=1}^{n} = f^{\text{epi}}_{\text{LLM}}(m). \tag{2}$$

Life Scene Memories. The three Life Scene Memories share a unified entry schema

$$M^{(i)}_{\text{scene}} = \big(\,\text{sid}_i,\ c_i,\ w_i,\ \mathbf{v}_i\,\big), \quad \mathbf{v}_i = f_{\text{emb}}(c_i), \tag{3}$$

where sid_i is a unique entry identifier within its life scene memory, scene $\in \{\text{Life}, \text{Work}, \text{Interest}\}$ indexes the three memories, c_i is an extracted trait content, $w_i \in [0, 1]$ its importance score assigned by the LLM, and $\mathbf{v}_i$ the embedding of the content.

3.4 Memory Retrieval

Given a user question q, CreaMem performs three operations: memory selection by a planning LLM, per-memory hybrid retrieval with balanced sampling, and global rank fusion.

Memory selection. The Planner LLM examines the question and selects a subset of relevant memories $R \subseteq \{\text{Core}, \text{Episodic}, \text{Life}, \text{Work}, \text{Interest}\}$ together with a keyword set $\mathcal{K}$ of 3 to 6 keywords used for retrieval:

$$(\mathcal{K},\ R) = f^{\text{plan}}_{\text{LLM}}(q). \tag{4}$$

Per-memory balanced retrieval. For each selected memory $r \in R$, two parallel retrieval paths are applied: BM25 over textual content and cosine similarity over the query embedding $\mathbf{v}_q = f_{\text{emb}}(\mathcal{K})$.

Each path returns the top-k candidates from r, denoted C_r^{bm} and C_r^{emb} respectively. Because every memory contributes the same number of candidates, no single memory dominates the pool. The combined candidate set C is formed by taking the union of all per-memory candidates and deduplicating by memory ID and content prefix:

$$C = \text{dedup}\Big(\bigcup_{r \in R} \big(C_r^{\text{bm}} \cup C_r^{\text{emb}}\big)\Big). \quad (5)$$

Reciprocal Rank Fusion. Candidates in C are re-ranked globally by both signals and fused via Reciprocal Rank Fusion (Cormack et al., 2009). For each $c \in C$,

$$\text{RRF}(c) = \frac{1}{k_0 + \text{rank}_{\text{bm}}(c)} + \frac{1}{k_0 + \text{rank}_{\text{emb}}(c)}, \quad (6)$$

where k_0 is a smoothing constant. The top-N candidates by RRF score are inserted into the prompt as retrieved context for response generation. If the Planner LLM judges the context insufficient, it selects memories to re-query at 3 entries each.

4 Experiments

4.1 Evaluation Setup

Datasets. We evaluate CreaMem on LoCoMo (Maharana et al., 2024) and LongMemEval-S (Wu et al., 2025). LoCoMo targets long-term conversational memory, with 10 ultra-long dialogues of around 300 turns and 9K tokens each, and 1,540 questions across four types: Single-hop, Multi-hop, Open-domain, and Temporal. LongMemEval-S contains 500 questions testing memory retention across multi-session dialogues.

Evaluation Metrics. We follow the MemGAS evaluation pipeline (Xu et al., 2026) for consistency with prior work. Following standard practice, our main metric is 4o-Judge accuracy, where GPT-4o serves as an LLM judge to assess response correctness against ground-truth answers. We additionally report Token F1 (Rajpurkar et al., 2016), BLEU-4, ROUGE-1/2/L, and BertScore for comprehensive lexical and semantic comparison.

Compared Methods. We compare CreaMem with representative methods covering the major paradigms of long-term conversational memory. To isolate the effect of memory architecture, all methods use GPT-4o-mini as their backbone LLM under an identical evaluation protocol.

Full History: This baseline places the entire dialogue history directly into the LLM context window without any retrieval or memory organization, serving as an unstructured upper reference.

MPNet (Song et al., 2020): A pretrained sentence encoder used as a dense retriever, indexing memory entries in a single, undifferentiated vector pool retrieved via cosine similarity.

Contriever (Izacard et al., 2022): An unsupervised dense retriever that produces general-purpose embeddings for similarity-based retrieval over an undifferentiated memory pool.

MPC (Lee et al., 2023): A prompted memory system that organizes conversational history through LLM-driven extraction without imposing explicit hierarchical or graph structure.

RecurSum (Wang et al., 2025): Recursively summarizes past dialogue segments into a hierarchical summary structure for long-term retention.

SeCom (Pan et al., 2025): Partitions dialogue into topical segments and retrieves at the segment granularity for coherent context recall.

RAPTOR (Sarthi et al., 2024): Builds a tree of recursive abstractive summaries that enables retrieval at multiple abstraction levels.

HippoRAG 2 (Gutiérrez et al., 2025): Constructs a knowledge graph over entities and relations extracted from past dialogues, and supports relational retrieval over the graph inspired by the hippocampal memory model.

A-Mem (Xu et al., 2025): An agentic memory system that maintains Zettelkasten-style interconnected notes with self-reflective links, enabling continuous memory evolution.

MemGAS (Xu et al., 2026): A multi-granularity memory system that maintains associations across abstraction levels for fine-grained retrieval.

MemoryOS (Kang et al., 2025): An OS-inspired memory system that partitions conversational memory along temporal scale into short-, mid-, and long-term tiers with heat-based promotion.

Notably, the chosen baselines are representative methods from each of the three paradigms identified in Section 2.1. Results for MemoryOS on LongMemEval-S are unavailable due to high runtime, as its memory organization requires a large number of LLM calls per turn, which we found impractical to reproduce on LongMemEval-S.

Implementation Details. We implement CreaMem with GPT-4o-mini as the backbone LLM for the Meta Memory Manager,

Model	4o-J	F1	B-4	R-1	R-2	R-L	BS	Avg. Tokens
LoCoMo								
Full History	33.43	12.23	1.84	12.70	5.66	11.73	84.07	20,078
MPNet (Song et al., 2020)	38.07	14.44	2.35	14.90	6.83	13.90	84.42	2,472
Contriever (Izacard et al., 2022)	40.33	15.66	2.67	16.01	7.68	15.00	84.65	2,348
MPC (Lee et al., 2023)	40.38	14.81	1.99	15.10	6.83	14.13	84.42	2,683
RecurSum (Wang et al., 2025)	22.56	9.14	0.99	9.82	3.38	8.98	83.45	3,074
SeCom (Pan et al., 2025)	44.21	13.79	2.30	14.28	6.17	13.30	84.04	**1,021**
HippoRAG 2 (Gutiérrez et al., 2025)	45.62	16.66	2.91	17.01	8.27	15.93	84.88	2,991
RAPTOR (Sarthi et al., 2024)	31.72	14.55	2.88	15.09	7.49	14.18	84.48	1,931
A-Mem (Xu et al., 2025)	40.81	14.72	2.83	16.22	7.71	14.89	84.72	3,042
MemGAS (Xu et al., 2026)	41.07	17.66	3.61	18.00	8.93	16.99	85.13	2,825
MemoryOS (Kang et al., 2025)	43.96	16.92	3.59	17.49	7.69	16.12	84.84	2,833
CreaMem (Ours)	**54.61**	**19.34**	**4.21**	**20.01**	**9.04**	**18.50**	**85.30**	2,805
LongMemEval-S								
Full History	50.60	11.48	1.40	12.10	5.47	10.85	83.07	103,137
MPNet (Song et al., 2020)	53.20	13.96	2.21	14.49	6.78	12.93	83.72	8,173
Contriever (Izacard et al., 2022)	55.40	13.78	2.21	14.46	6.93	12.89	83.70	8,286
MPC (Lee et al., 2023)	53.80	13.60	1.74	14.27	6.49	12.95	83.49	8,457
RecurSum (Wang et al., 2025)	35.40	12.29	2.09	13.01	5.55	11.52	83.60	8,853
SeCom (Pan et al., 2025)	56.00	12.95	2.25	13.80	6.09	11.93	83.51	**2,741**
HippoRAG 2 (Gutiérrez et al., 2025)	57.60	14.73	2.15	15.30	7.36	13.83	83.86	8,530
RAPTOR (Sarthi et al., 2024)	32.20	12.08	1.90	12.73	5.82	11.25	83.50	6,254
A-Mem (Xu et al., 2025)	55.60	13.73	2.11	14.82	6.81	12.98	83.88	9,018
MemGAS (Xu et al., 2026)	60.20	20.38	4.22	21.05	10.47	19.47	85.21	8,829
CreaMem (Ours)	**66.40**	**20.71**	**4.79**	**21.82**	**11.13**	**19.78**	**85.36**	7,250

Table 1: Main results on LoCoMo (top) and LongMemEval-S (bottom). Best per column in bold.

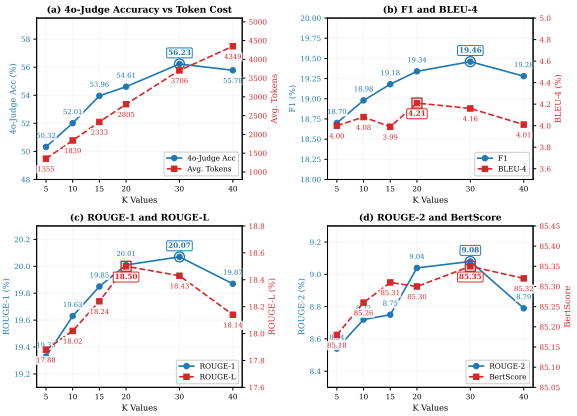

Figure 2: Sensitivity of retrieval budget k on LoCoMo. (a) 4o-Judge accuracy and token cost. (b) F1, BLEU-4. (c) ROUGE-1/L. (d) ROUGE-2, BertScore.

all memory-specific extractors, and the LLM Planner, with temperature set to 0 throughout. Memory is persisted in a local SQLite database, and text embeddings are produced by `text-embedding-3-small`. Batch extraction triggers once 3 or more user messages accumulate.

At retrieval, the Planner LLM outputs 3 to 6 keywords per query; per-memory top-k and final top-N are 20 in LoCoMo and 40 in LongMemEval-S. Figure 2 shows that k=30 yields marginally higher accuracy in LoCoMo at the cost of substantially more retrieved tokens. We deliberately adopt the lower k=20 to match the token budget of the baselines in Table 1, ensuring that the comparison isolates memory organization from context size. The RRF smoothing constant k_0 is 60 following standard practice (Cormack et al., 2009). Core Memory has a character limit of $L = 2000$ with automatic compression at 90% capacity.

For evaluation, GPT-4o serves as the LLM judge for the 4o-Judge metric. We adopt the QA and judge prompts of MemGAS (Xu et al., 2026) for fair comparison with prior work. Full prompt templates are in Appendix B.

4.2 Main Results

Table 1 presents the aggregate results on LoCoMo and LongMemEval-S. We have the following observations:

(1) **Flat-storage methods lack organizational structure for retrieval.** MPNet, Contriever, and MPC index all memory entries in a single undifferentiated pool. Their accuracy forms a stable middle tier on both benchmarks, indicating that pure similarity-based retrieval has a clear ceiling without further organization.

(2) **Structure-imposing methods narrow but**

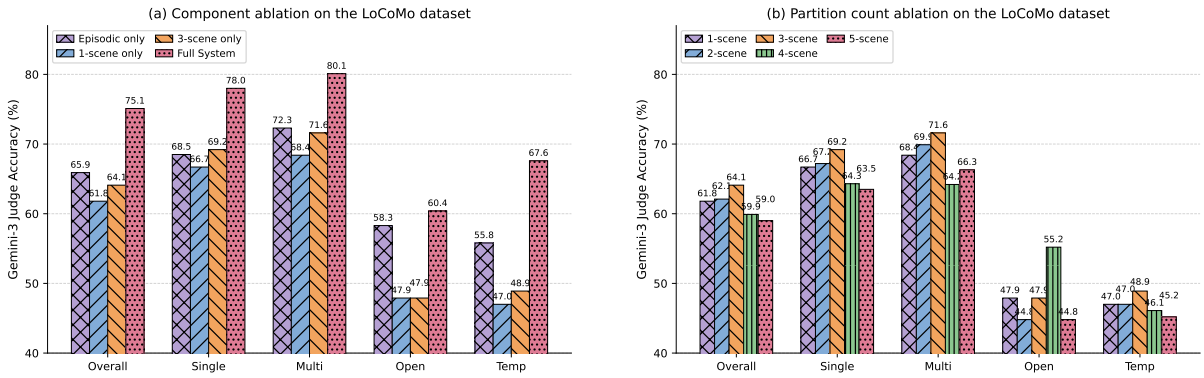

Figure 3: Ablation results on the LoCoMo dataset. (a) Component ablation. (b) Partition count ablation.

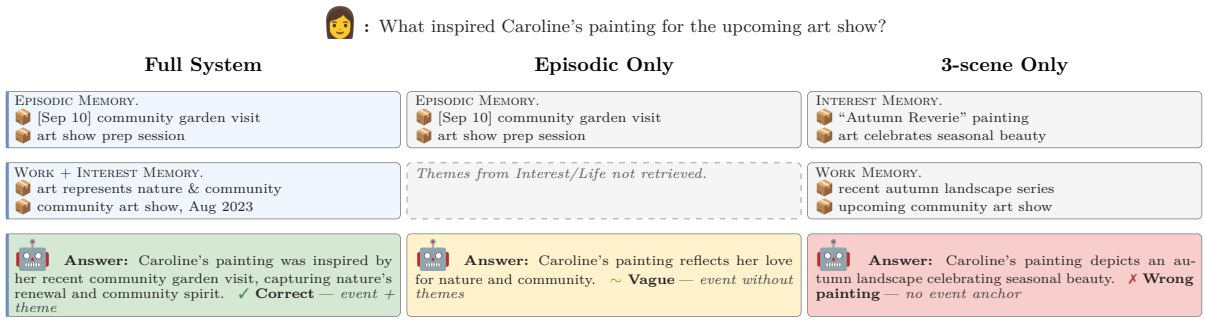

Figure 4: Case study on a multi-hop LoCoMo question. The Full System integrates episodic and scene-level evidence, while either ablation produces incomplete or temporally ungrounded answers.

do not close the gap. HippoRAG 2 and MemGAS lead this group on LoCoMo and LongMemEval-S respectively, while summary-based variants such as RecurSum and RAPTOR trade factual specificity for compactness and form the lowest tier. The shared ceiling across this group reflects a common limitation: structure is imposed within a single, undifferentiated memory pool, so entries from unrelated life scenes still compete for the same retrieval space, producing cross-scene interference.

(3) **Memory-partitioning by temporal scale does not relieve cross-scene interference.** MemoryOS partitions memory along temporal scale rather than along user's life scenes, and on LoCoMo lands in the same performance band as the strongest structure-imposing baselines. However, entries from unrelated life scenes can still co-occur within the same tier, leaving cross-scene interference unresolved at retrieval time.

(4) **CreaMem achieves the best performance across both datasets and all metrics.** It surpasses HippoRAG 2 and MemGAS, the strongest baselines on LoCoMo and LongMemEval-S respectively, and the lead is consistent across F1, BLEU-4, ROUGE-1/2/L, and BertScore rather than concentrated on a single metric. This breadth indicates that the gain comes from selecting more relevant content, not from surface-level lexical overlap.

(5) **The gains come from organization, not from a larger context budget.** CreaMem's token consumption is on par with the strongest structured baselines on LoCoMo and below most baselines on LongMemEval-S, where it uses an order of magnitude fewer tokens than Full History, yet it delivers the highest accuracy. Improvement therefore arises from how memory is organized and retrieved, not from feeding the LLM more raw context.

4.3 Ablation Study

We conduct three ablations on the full LoCoMo corpus using Gemini-3 as both QA backbone and judge. Since architectural benefits on an early-generation backbone like GPT-4o-mini do not automatically transfer to newer LLMs with stronger reasoning and context handling, we re-run all configurations on Gemini-3 to verify that CreaMem's gains persist on a substantially newer backbone. Across all configurations, the retrieval budget is fixed at 20 memory entries per query and the underlying memory entries are identical: ablated variants

reorganize the same extracted content rather than re-extracting. Core Memory and the re-query step are held out across all variants, as both are orthogonal to the scene-partitioning axis under study. Observed differences therefore reflect how memory is structured at retrieval, not how much information is stored or surfaced into context.

Component Ablation. Figure 3(a) compares four configurations testing whether CreaMem's two design choices, scene partitioning and dual encoding, are necessary or redundant.

First, we test whether three scene memories add value beyond a single undifferentiated pool, since partitioning could simply complicate retrieval. We compare **3-scene only**, which retains the three scene memories without Episodic Memory, with **1-scene only**, which merges them into a single pool while keeping every trait entry verbatim. Under identical content and retrieval budget, 3-scene outperforms 1-scene by 2.3 points, showing that scene-based organization is not redundant against flat storage at retrieval time.

Second, we test whether storing each experience as both an episodic event and a trait merely duplicates information. We compare **Full System**, which couples Episodic Memory with three scene memories, against **Episodic only**, which retains timeline and event details alone. Full System reaches 75.1% overall, 9.2 points above Episodic only, with the gap widening to over 11 points on Temporal questions. This gain is not from feeding more context: at the same retrieval budget, Episodic only consumes 42% more tokens than Full System, since each episodic entry's details field preserves the full event narrative and surrounding context, carrying even more information per entry than Full System. Dual encoding therefore improves accuracy while reducing token consumption. Figure 4 illustrates this on a multi-hop case where ablating either side yields incomplete or temporally ungrounded answers.

Third, we test whether finer partitioning further improves performance. Figure 3(b) compares five partition counts under a fixed retrieval budget of 20 entries per query, evenly distributed across memories, matching the baseline cost in Table 1. **1-scene** merges all scenes, **2-scene** merges Life and Work, **3-scene** is our default, **4-scene** splits Life into "family"/"other", and **5-scene** further splits Interest into "hobby"/"entertainment", with splits produced by LLM re-tagging existing entries. Among the partition counts tested under this retrieval budget, the 3-scene Life/Work/Interest configuration achieves the best balance: coarser partitions force unrelated scene content to compete for the same retrieval slots, while finer partitions over-fragment memory and dilute per-memory information density.

Judge	Coverage (%)
GPT-4o	71.00
Human Rater A	83.00
Human Rater B	90.10
Human Rater C	80.50
LLM–Human agreement	77.20

Table 2: Coverage of the Life/Work/Interest taxonomy over 1,000 sampled LoCoMo dialogue turns, 100 per conversation, judged by GPT-4o and three human raters.

4.4 Hyperparameter Analysis

We analyze the impact of the per-memory retrieval budget k on model performance. As shown in Figure 2, accuracy is non-monotone in k: 4o-Judge peaks at k=30 and declines at k=40, with the same pattern across F1, ROUGE-1/2/L, and BertScore, while token cost grows nearly linearly. Beyond a modest budget, additional entries act as noise rather than signal, degrading answers despite higher cost.

4.5 Architectural Validation

CreaMem rests on the prerequisite that the Life/Work/Interest taxonomy covers the bulk of personal memory content. We validate this empirically by sampling 1,000 dialogue turns from the 10 LoCoMo conversations, 100 turns per conversation. Each turn is independently judged by GPT-4o and three human raters on whether it falls within one of the three scenes. As shown in Table 2, coverage is consistently high across judges, with strong LLM–human agreement. The remaining gap comes from turns that are hard to assign to any scene, such as greetings or small talk. Overall, most personal content in long-term dialogue fits into the three scenes validating CreaMem's design prerequisite.

4.6 Failure Analysis

Aggregate improvement does not imply per-question dominance: CreaMem still fails on some questions answered correctly by competing methods. In a Single-hop question about Nate's second tournament, CreaMem stored the event and the game name, *Street Fighter*, in separate entries. Only the event entry appeared in the retrieved context, whereas SeCom retained the contiguous dia-

logue segment and answered correctly. In an Open-domain question about John's endorsement deal, CreaMem retrieved both the unnamed outdoor-gear deal and his earlier interest in Under Armour. Response generation nevertheless failed to connect the entries, whereas MemoryOS produced the reference answer. The first case reflects incomplete retrieval of linked entries; the second reflects failure to combine retrieved entries after retrieval. These cases motivate links to underlying dialogue turns, fallback to relevant dialogue segments, and explicit evidence combination. Appendix A summarizes the source evidence, model outputs, and one additional retrieval failure.

5 Conclusion

We presented CreaMem, a scene-aware memory architecture motivated by autobiographical memory organization and dual coding in cognitive psychology. It partitions memory along the user's life scenes alongside Episodic and Core Memory, dual-encoding each experience and combining views via per-memory balanced sampling. Experiments on LoCoMo and LongMemEval-S show consistent gains in accuracy and multi-hop reasoning at comparable token cost, suggesting that the key lever for coherent long-term agents is how memory is organized, not how much context is supplied.

Limitations

CreaMem has several limitations. First, the fixed Life/Work/Interest taxonomy is a deliberately coarse organization rather than a universal or optimal ontology. Although our results support this partition on the evaluated benchmarks, it may not transfer uniformly across users, cultures, or specialized domains. Future work will investigate configurable or automatically induced scene partitions that adapt to individual users while retaining interpretable routing.

Second, CreaMem incurs additional system overhead. End-to-end latency averages 2.9 seconds per query, compared with 1.98 seconds for A-Mem. The sufficiency check can also trigger a second retrieval round, improving recall at the cost of additional latency and tokens. Query-adaptive pruning, asynchronous retrieval, and learned stopping criteria may reduce this overhead.

Several evaluation challenges remain for LLM-extraction memory systems. Write-time costs are difficult to compare because methods differ in batching granularity, module prompts, and the abstraction level of stored entries. Retrieval metrics such as Recall@k and NDCG@k also assume direct alignment between retrieved items and reference evidence. This assumption does not hold when memory entries summarize or combine multiple dialogue turns. Finally, extraction behavior depends on prompt and model choice. Standardized write-time accounting, source-aligned retrieval evaluation, and method-agnostic extraction protocols remain important directions for future work.

Acknowledgments

This work is supported by the Early Career Scheme (No.CityU 21219323) and the General Research Fund (No.CityU 11220324) of the University Grants Committee (UGC), the NSFC Young Scientists Fund (No.9240127), and the Donation for Research Projects (No.9229216).

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.

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

Martin A Conway. 2005. Memory and the self. *Journal of Memory and Language*, 53(4):594–628.

Martin A Conway and Christopher W Pleydell-Pearce. 2000. The construction of autobiographical memories in the self-memory system. *Psychological Review*, 107(2):261–288.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 758–759.

Junyu Feng, Binxiao Xu, Jiayi Chen, Mengyu Dai, Cenyang Wu, Haodong Li, Bohan Zeng, Yunliu Xie, Hao Liang, Ming Lu, and Wentao Zhang. 2026. M2A: Multimodal memory agent with dual-layer hybrid memory for long-term personalized interactions. *arXiv preprint arXiv:2602.07624*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.

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

William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30.

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

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. 2025. Memory OS of AI agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25961–25970.

Gibbeum Lee, Volker Hartmann, Jongho Park, Dimitris Papailiopoulos, and Kangwook Lee. 2023. Prompted LLMs as chatbot modules for long open-domain conversation. In *Findings of the Association for Computational Linguistics: ACL 2023*.

Mingcong Lei, Honghao Cai, Yuyuan Yang, Yimou Wu, Jinke Ren, Zezhou Cui, Liangchen Tan, Junkun Hong, Gehan Hu, Shuangyu Zhu, Shaohan Jiang, Ge Wang, Junyuan Tan, Zhenglin Wan, Zheng Li, Zhen Li, Shuguang Cui, Yiming Zhao, and Yatong Han. 2026. RoboMemory: A brain-inspired multi-memory agentic framework for interactive environmental learning in physical embodied systems. *arXiv preprint arXiv:2508.01415*.

Patrick 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. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Yang Li, Jiaxiang Liu, Yusong Wang, Yujie Wu, and Mingkun Xu. 2026. BMAM: Brain-inspired multi-agent memory framework. *arXiv preprint arXiv:2601.20465*.

Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, Rui Kong, Yile Wang, Hanfei Geng, Jian Luan, Xuefeng Jin, Zilong Ye, Guanjing Xiong, Fan Zhang, Xiang Li, and 6 others. 2024. Personal LLM agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459*.

Minhua Lin, Zhiwei Zhang, Hanqing Lu, Hui Liu, Xianfeng Tang, Qi He, Xiang Zhang, and Suhang Wang. 2026. MemMA: Coordinating the memory cycle through multi-agent reasoning and in-situ self-evolution. *arXiv preprint arXiv:2603.18718*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

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

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*. ArXiv preprint arXiv:2402.17753.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Jianfeng Gao. 2025. On memory construction and retrieval for personalized conversational agents. In *The Thirteenth International Conference on Learning Representations*. ArXiv:2502.05589.

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. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*.

Norman Paulsen. 2025. Context is what you need: The maximum effective context window for real world limits of LLMs. *arXiv preprint arXiv:2509.21361*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

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

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations (ICLR)*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33.

Sunil Tiwari and Payal Fofadiya. 2026. Multi-layered memory architectures for LLM agents: An experimental evaluation of long-term context retention. *arXiv preprint arXiv:2603.29194*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Endel Tulving. 1972. Episodic and semantic memory. *Organization of Memory*, pages 381–403.

Endel Tulving. 2002. Episodic memory: From mind to brain. *Annual Review of Psychology*, 53:1–25.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Qingyue Wang, Yanhe Fu, Yanan Cao, Shuai Wang, Zhiliang Tian, and Liang Ding. 2025. Recursively summarizing enables long-term dialogue memory in large language models. *Neurocomputing*. ArXiv:2308.15022.

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

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. LongMemEval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*.

Yanchen Wu, Tenghui Lin, Yingli Zhou, Fangyuan Zhang, Qintian Guo, Xun Zhou, Sibo Wang, Xilin Liu, Yuchi Ma, and Yixiang Fang. 2026. Memory in the LLM era: Modular architectures and strategies in a unified framework. *arXiv preprint arXiv:2604.01707*.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Wenlin Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, and Tong Xu. 2026. From single to multi-granularity: Toward long-term memory association and selection of conversational agents. In *International Conference on Learning Representations (ICLR)*. ArXiv preprint arXiv:2505.19549.

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

Ke Yang, Zixi Chen, Xuan He, Jize Jiang, Michel Galley, Chenglong Wang, Jianfeng Gao, Jiawei Han, and ChengXiang Zhai. 2026. PlugMem: A task-agnostic plugin memory module for LLM agents. *arXiv preprint arXiv:2603.03296*.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. 2026. MemAgent: Reshaping long-context LLM with multi-conv RL-based memory agent. *arXiv preprint arXiv:2507.02259*.

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

Yanfang Zhou, Xiaodong Li, Yuntao Liu, Yongqiang Zhao, Xintong Wang, Zhenyu Li, Jinlong Tian, and Xinhai Xu. 2025. M2PA: A multi-memory planning agent for open worlds inspired by cognitive theory. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 23204–23220.

A Supplementary Analyses

Category-level and open-weight evaluation. Under the matched GPT-4o-mini/GPT-4o protocol, CreaMem leads Overall, Multi-hop, and Temporal accuracy, whereas SeCom and MemoryOS lead Single-hop and Open-domain, respectively

(Table 3a). Its margins over the strongest baseline are 2.83 points on Multi-hop and 19.93 points on Temporal questions. With Qwen3.6-35B-A3B, CreaMem leads by 5.97 points overall and remains strongest on Multi-hop, Open-domain, and Temporal questions (Table 3b), supporting cross-backbone transfer but not uniform dominance.

Routing, extraction, and taxonomy reliability. Across 998 jointly judged LoCoMo turns, routing reaches 67.94% exact match, 94.79% partial match, and 74.53/73.65 Micro/Macro-F1 (Table 4). At 40% synthetic misrouting, QA declines only 1.23 points, from 54.61% to 53.38%. In a 500-memory audit, Gemini 3.1 Pro/GPT-4o find 90.40/93.60% of entries retain a supported central fact, while 9.60/6.40% contain an unsupported central claim. Among 107 vertical-domain LongMemEval-S turns, exact routing is 68.22% and at-least-one-scene coverage is 99.03%. Coverage is therefore broad, although routing and extraction errors remain.

Allocation and architectural ablations. Dynamic allocation reaches 55.26% versus 55.00% for balanced sampling ($+0.26$ points; 95% CI $[-0.78, 1.30]$) but adds 0.69 LLM calls per question. Under the matched GPT-4o-mini/GPT-4o setup, Full CreaMem reaches 53.24%, versus 44.03% for Episodic-only and 40.32–41.43% for scene-only variants. RRF changes overall accuracy by only $+0.06$ and -0.91 points across two backbones, with both confidence intervals spanning zero. The main gain therefore arises from scene-aware dual encoding rather than allocation or fusion alone.

Representative failure cases. Table 5 distinguishes failures to retrieve linked or implicit evidence from failures to combine evidence already present in the context.

B Prompt Templates

For brevity, we list only the key prompts of CreaMem below; additional prompts are documented in the released code. Prompts A.1–A.4 cover CreaMem's routing and extraction logic and are abridged for readability. Prompts A.5 and A.6 reproduce the QA and judge prompts from MemGAS (Xu et al., 2026) for completeness.

Meta Memory Manager (Storage Routing)

Role. You are the Meta Memory Manager of CreaMem, responsible for routing batched user messages to the appropriate memory components.
Task. Analyze a batch of user messages and call `trigger_memory_update(memory_types=[...])` with one or more types from `{core, episodic, life, work, interest}`, then call `finish_memory_update()`.
Memory definitions.

- `core`: WHO the user is and HOW to interact (identity, preferences, relationships, static reference data).
- `episodic`: WHAT happened WHEN (time-anchored events and activities).
- `life`: daily life, health, family, social, food, shopping.
- `work`: professional or academic activities, projects, meetings, learning.
- `interest`: hobbies, entertainment, leisure.

Routing rules.

- Include `episodic` for nearly all messages containing events.
- Include `life` / `work` / `interest` only when relevant content is present.
- Include `core` only when the message reveals new fundamental user information.
- When uncertain between scenes, prefer including more rather than missing relevant content.

Example.
Input: "Went to the gym with my colleague after work."
Output: `trigger_memory_update([episodic, work, life])`

Figure 5: Routing prompt used by the Meta Memory Manager at storage time.

Planner LLM

Role. You are the retrieval planner of CreaMem. Given a user question, select which memory components to search and the keywords to use.
Task. Output a JSON object with two fields:

- `keywords`: 3 to 6 keywords extracted from the question, used for BM25 and embedding retrieval.
- `memory components`: a subset of `{episodic, life, work, interest}` indicating which components to query. Include the entire Core Memory if you think answering the question requires the user's profile.

Selection guidance.

- Questions about specific events or timestamps → include `episodic`.
- Questions about user traits, habits, or preferences → include the relevant Life Scene Memories.
- Multi-aspect questions → select multiple components

Fallback. On parse failure or empty output, the system retrieves from all four components with the question itself as the keyword.
Example.
Question: "When did the user go camping with family?"
Output: `{"keywords": ["camping", "family", "trip"], "components": ["episodic", "life"]}`

(a) GPT-4o-mini backbone and GPT-4o judge.

Method	Overall	Single-hop	Multi-hop	Open-domain	Temporal
CreaMem	**54.61**	66.11	**28.72**	38.54	**52.02**
HippoRAG 2	47.66	63.73	24.11	35.42	29.91
SeCom	45.58	**67.66**	23.40	34.38	10.59
MemoryOS	43.96	54.70	25.89	**42.71**	32.09
MemGAS	43.38	57.19	23.05	33.33	28.04

(b) Qwen3.6-35B-A3B backbone with thinking disabled.

Method	Overall	Single-hop	Multi-hop	Open-domain	Temporal	Avg. Tokens
CreaMem	**57.66**	66.11	**31.21**	**39.58**	**64.17**	2,915.64
HippoRAG 2	51.69	64.21	29.79	29.17	44.86	2,482.88
SeCom	47.01	**69.56**	26.60	35.42	9.35	907.19
MemoryOS	39.35	50.42	19.86	36.46	28.35	3,087.18
MemGAS	37.79	46.49	18.79	35.41	32.40	3,214.00

Table 3: Category-level LoCoMo results under closed and open-weight backbones. Values are 4o-Judge accuracy (%); average QA tokens are reported for the Qwen evaluation. Best results within each backbone are bold.

Analysis	Protocol	Result
Routing quality	998 jointly judged LoCoMo turns	Exact 67.94%; partial 94.79%; Micro/Macro-F1 74.53/73.65%.
Synthetic misrouting	1,540 questions; Work → Interest	Accuracy at 0/10/20/40% corruption: 54.61/54.61/54.29/53.38%.
Extraction faithfulness	500 memories; Gemini 3.1 Pro/GPT-4o judges	Supported central fact: 90.40/93.60%; fully supported: 71.20/73.40%; unsupported central claim: 9.60/6.40%.
Scene taxonomy	107 vertical-domain LongMemEval-S turns	Conservative judge intersection: 68.22% exact routing and 99.03% at-least-one-scene coverage among non-empty references.
Retrieval allocation	1,540 paired LoCoMo questions	Balanced 55.00% versus Dynamic 55.26%; $\Delta = +0.26$ pp, 95% CI $[-0.78, 1.30]$; Dynamic adds 0.69 LLM calls/question.
Components	GPT-4o-mini/GPT-4o; no Core Memory	Full System 53.24%; Episodic-only 44.03%; scene-only variants 40.32–41.43%.
RRF vs. BM25-only	Gemini-3 and GPT-4o-mini/GPT-4o	$+0.06$ pp, 95% CI $[-2.01, 2.08]$; and -0.91 pp, 95% CI $[-2.99, 1.10]$, respectively.

Table 4: Reliability audits and supplementary ablations. Routing, misrouting, allocation, component, and RRF results use LoCoMo; the scene-taxonomy audit uses LongMemEval-S. Values follow the rebuttal protocols.

Question (reference)	CreaMem behavior	Correct baseline and diagnosis
Single-hop: Nate's second tournament (*Street Fighter*)	The event and game were stored separately, but only the event entry was retrieved; the response stated that the game was unknown.	SeCom retained D10:4–D10:6 and answered correctly. Linked entries were not retrieved together.
Open-domain: John's outdoor-gear endorsement (*Under Armour*)	Both the unnamed deal and John's earlier interest in Under Armour were retrieved, but the response did not connect them.	MemoryOS answered correctly. Retrieved evidence was not combined during response generation.
Temporal: Holiday season of Evan's wedding (*Christmas*)	The date-bearing wedding session was absent from the context, so the response stated that the holiday could not be determined.	MemoryOS answered correctly. Implicit temporal evidence was not retrieved.

Table 5: Representative LoCoMo questions answered incorrectly by CreaMem but correctly by a baseline. The cases illustrate distinct failure stages rather than their prevalence.

Figure 6: Planner prompt that produces (K, R) at retrieval time.

Episodic Memory Extractor

Role. You are the Episodic Memory Manager. You receive batched user messages with timestamps and extract distinct events as separate entries.
Schema. Each episodic entry is a tuple `(event_type, summary, details, actor)` with timestamp inherited from the conversation.
Extraction rules.

- Identify all distinct events in the batch; each event becomes a separate entry.
- Convert relative time references ("yesterday", "last week") to absolute dates using the conversation timestamp.
- Preserve proper nouns, numbers, titles, and concrete details verbatim.
- Skip greetings, small talk, and content already present in existing entries.

Tool calls. Use `episodic_memory_insert(items=[...])` by default. Use `episodic_memory_merge` only to continue the exact same event within one turn. Make exactly one tool call per batch.
Example.
Input (timestamp 2023-07-06): "We went camping at the beach last weekend, and I painted a sunset last year."
Output:

```
episodic_memory_insert([
  {event_type: activity,
   summary: "User went camping at the
            beach around July 1-2, 2023",
   details: "Camped at the beach with
             family on 2023-07-01 to 07-02",
   actor: user},
  {event_type: activity,
   summary: "User painted a sunset in 2022",
   details: "User painted a sunset in 2022",
   actor: user}
])
```

Figure 7: Extraction prompt for the Episodic Memory.

Scene Memory Extractor (Life / Work / Interest)

The three Life Scene Memories share a common extractor template. Only the scene definition (highlighted below) differs across instantiations.
Role. You are the {**SCENE**} Memory Manager of CreaMem. You extract scene-specific traits from batched user messages.
Scene definitions.

- `life`: health, family, daily routines, food preferences, social interactions, shopping, medical.
- `work`: current role, projects, professional skills, deadlines, learning, career goals.
- `interest`: hobbies, entertainment, sports, music, movies, travel, creative pursuits.

Schema. Each scene entry is a tuple (`content`, `importance_score`), where `content` is a trait, and `importance_score` $\in [0, 1]$.
Extraction rules.

- Extract distilled trait inferences, not raw events.
- Each distinct fact becomes a separate entry.
- Preserve proper nouns, numbers, and concrete details verbatim.
- Ignore content outside the assigned scene.

Tool calls. Use `{scene}_memory_insert(items=[...])` by default. Use `{scene}_memory_update` only when an existing entry's information has genuinely changed.
Example (Life).
Input: "I'm single, moved from Sweden 4 years ago, and have 3 pets: Oliver (dog), Luna (cat), Bailey (dog)."
Output:

```
life_memory_insert([
  {content: "User is single",
   score: 0.7},
  {content: "User moved from Sweden in 2019",
   score: 0.8},
  {content: "User has 3 pets: Oliver (dog),
            Luna (cat), Bailey (dog)",
   score: 0.7}
])
```

Figure 8: Shared extraction template used by the three Life Scene Memories. Only the scene definition varies across instantiations.

QA Prompt

You are an intelligent dialog bot. You will be shown History Dialogs. Please read, memorize, and understand the given Dialogs, then generate one concise, coherent and helpful response for the Question.

```
History Dialogs: {retrieved_texts}
Question Date: {question_date}
Question: {question}
```

Figure 9: QA prompt used to generate responses, identical to that of MemGAS (Xu et al., 2026) (following Lu et al., 2023; Pan et al., 2025).

GPT-4o Judge Prompt

I will give you a question, a reference answer, and a response from a model. Please answer `[[yes]]` if the response contains the reference answer. Otherwise, answer `[[no]]`. If the response is equivalent to the correct answer or contains all the intermediate steps to get the reference answer, you should also answer `[[yes]]`. If the response only contains a subset of the information required by the answer, answer `[[no]]`.

```
[User Question]
{question}
[The Start of Reference Answer]
{answer}
[The End of Reference Answer]
[The Start of Model's Response]
{response}
[The End of Model's Response]
```

Is the model response correct? Answer `[[yes]]` or `[[no]]` only.

Figure 10: GPT-4o judge prompt used to score response correctness against ground truth, identical to that of MemGAS (Xu et al., 2026).