CHOOSING HOW TO REMEMBER: ADAPTIVE MEMORY STRUCTURES FOR LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Memory is critical for enabling large language model (LLM)-based agents to maintain coherent behavior over long-horizon interactions. However, existing agent memory systems suffer from two key gaps: they rely on a one-size-fits-all memory structure and do not model memory structure selection as a context-adaptive decision, limiting their ability to handle heterogeneous interaction patterns and resulting in suboptimal performance. We propose a unified framework, FLUXMEM, that enables adaptive memory organization for LLM agents. Our framework equips agents with multiple complementary memory structures, including linear, graph, and hierarchical. It explicitly learns to select among these structures based on interaction-level features, using offline supervision derived from downstream response quality and memory utilization. To support robust long-horizon memory evolution, we further introduce a three-level memory hierarchy and a Beta Mixture Model-based probabilistic gate for distribution-aware memory fusion, using posterior probabilities to adapt fusion decisions to the matching-score distribution. Experiments on two long-horizon benchmarks, PERSONAMEM and LoCoMo, demonstrate improvements of 8.85 percentage points in average accuracy and 5.28 points averaged across F1, BLEU-1, and ROUGE-L, respectively, over the strongest baseline for each metric.

1 INTRODUCTION

Large language model (LLM)-based agents have shown strong performance in interactive settings, including multi-step planning and open-ended dialogue (Schlegel et al., 2025; Wang et al., 2024; Li et al., 2024; 2025a; Jia et al., 2025). In long-horizon interactions, agents must accumulate, retain, and retrieve information across many turns while user goals and contexts evolve over time (Maharana et al., 2024; Wu et al., 2025a). In this process, agent memory is responsible for recalling relevant information, such as preferences, from rich, multi-turn user interactions. It directly influences the coherence of agents' responses and long-term behavioral consistency. Therefore, effective memory organization is essential (Zhang et al., 2025a; Pan et al., 2025).

Prior work on long-horizon memory for LLM agents falls into three lines. (1) *Flat retrieval-based memory systems* store salient facts or summaries in weakly structured stores and retrieve them via similarity search, offering efficiency but limited abstraction and relational reasoning (Chhikara et al., 2025; Fang et al., 2025; Zhong et al., 2024; Gutiérrez et al., 2025; Qian et al., 2025). (2) *Explicitly structured memory systems* organize memory into linked notes or graph-like representations, enabling richer relational and temporal reasoning but typically relying on fixed organizational schemes (Xu et al., 2025a;b; Wang et al., 2025; Wang & Chen, 2025; Anonymous, 2026; Rasmussen et al., 2025; Liu et al., 2023). (3) *Policy managed memory systems* regulate storage and retrieval through higher-level control mechanisms. They are either learned (e.g., reinforcement learning-based controllers) or engineered (e.g., OS-inspired designs like decay and redundancy filtering) (Yan et al., 2025; Agrawal et al., 2025; Kang et al., 2025; Jonelagadda et al., 2025; LangChain, 2025; Du et al., 2025; Li et al., 2025b; Packer et al., 2023; Asai et al., 2024; Shinn et al., 2023).

As illustrated in Fig. 1, existing methods often struggle to deliver accurate and consistent responses in complex, long-horizon interaction scenarios, due to the following two research gaps. **Gap 1: Single-structure assumption within memory systems.** Most existing methods (Kang et al., 2025; Rasmussen et al., 2025) assume that a single memory structure (e.g., graph) is sufficient across

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.

various tasks. Here, a memory structure refers to how the content within an individual memory unit (i.e., a compact dialogue segment) is organized and accessed, whereas relationships between different memory units concern how separate memories are linked or associated with one another. This one-size-fits-all memory selection leads to suboptimal performance, as different tasks exhibit distinct structural characteristics and interaction patterns (Trach & McDougle, 2025; Cellier et al., 2022; Zhang et al., 2025b), such as topic evolution, temporal progression, and relational dependencies, that often co-occur in long-horizon conversations. **Gap 2: Lack of conversation-adaptive memory structure selection.** While Gap 1 concerns how memory structure is designed, Gap 2 focuses on how memory structure is used during interaction. Although some methods employ different memory structures across memory layers (e.g., linear for short-term and graph for long-term) (Wang et al., 2025), each layer remains fixed and non-adaptive to interaction content. As a result, structure choice is excluded from optimization, limiting alignment with downstream performance and robustness under distribution shifts across users and tasks.

To bridge these two gaps, we propose a novel framework, FLUXMEM, that dynamically selects memory structures for LLM agents. Existing memory systems share a blind spot: **Memory structure stays fixed across interactions.**

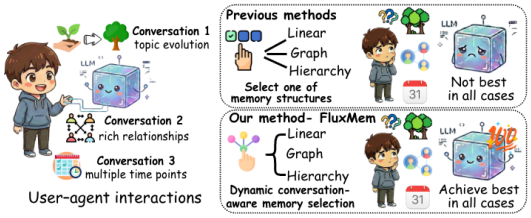

Figure 1: User-agent interactions exhibit diverse structural patterns, including relational, temporal, and topical variations. Prior methods rely on a single memory structure and struggle across interaction types. Our approach dynamically selects memory structures based on the conversation, enabling robust performance.

To tackle Gap 1, we move beyond the assumption of a single universal memory structure and equip agents with multiple complementary memory structures, each inducing distinct retrieval behaviors over the same interaction history. This expands the expressiveness of agent memory and enables structural flexibility across heterogeneous tasks. **To overcome Gap 2**, we elevate memory structure selection to an explicit, learnable mechanism. We optimize the structure selection decision using offline, interaction-derived feedback obtained from the Multi-Session Chat (MSC) dataset (Xu et al., 2022), aligning memory structures with downstream response quality and retrieval effectiveness. In addition, we address a common practice in memory systems: using fixed similarity thresholds to decide whether a newly retrieved memory should be fused with an existing one. These thresholds are usually set by hand and assume that similarity scores follow a stable pattern, which makes them sensitive to noise and unreliable when interaction distributions change. Instead, we treat memory integration as a probabilistic gating process and propose a new Beta Mixture Model (BMM)-based gate that models the distribution of similarity scores directly. By fitting the matching-score distribution, our approach replaces a fixed raw-similarity cutoff with a posterior decision rule controlled by τ_{BMM}. On LoCoMo and PERSONAMEM, our approach consistently improves response accuracy and cross-turn consistency over existing memory systems. Our contributions follow:

❶ **Key Observations.** We identify fixed structures and missing selection feedback as two gaps.

❷ **Adaptive Multi-Structure Memory Framework.** We unify complementary memory structures with selection learned from interaction-derived feedback.

❸ **Memory Fusion and Update Mechanisms.** We introduce mechanisms for memory fusion and update, including a BMM-based probabilistic gate.

2 RELATED WORK

LLMs have been widely applied across diverse domains, including dialogue systems, decision-making agents, code generation, and scientific reasoning (Yang et al., 2025; Li et al., 2025a; Jia et al., 2025; Chen et al., 2026; Wu et al., 2025b; Lu et al., 2025; Jiang et al., 2025b). Their in-context learning and generative capabilities make them a foundation for autonomous agent systems that require long-horizon interaction and memory management. Long-term LLM memory research spans three paradigms: flat retrieval-based, explicitly structured, and policy-managed systems.

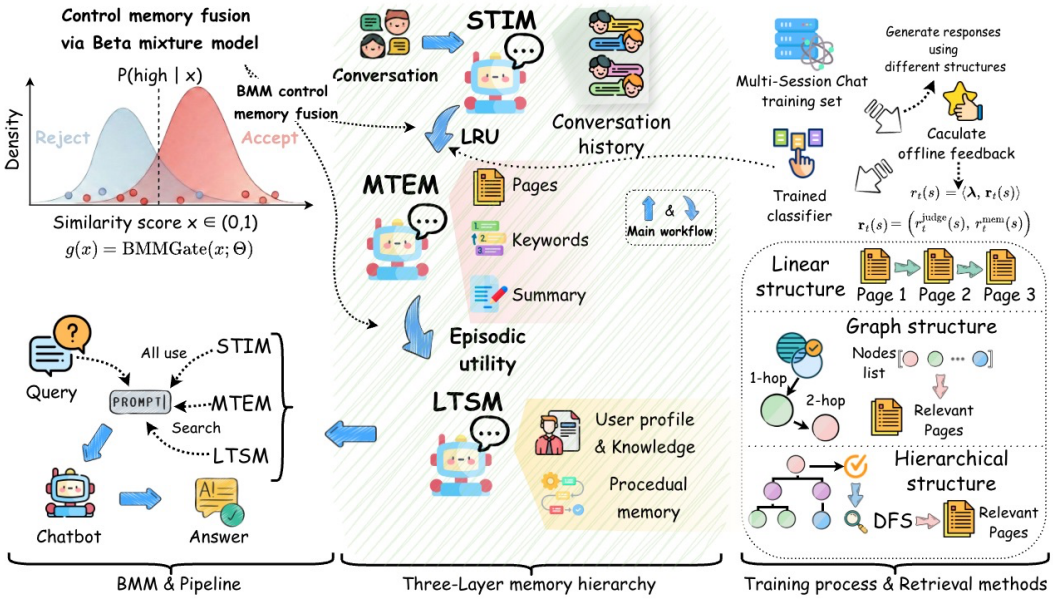

Figure 2: Overall architecture of *FluxMem*. The framework consists of three memory layers: STIM for buffering recent context, MTEM for structure-aware episodic storage, and LTSM for long-term semantic consolidation. The central part depicts the **main workflow**, including memory writing, structure selection, and query-time retrieval across memory layers.

Flat retrieval-based memory systems store facts, events, or summaries in flat repositories for similarity-based recall. Mem0, LightMem, and MemoryBank emphasize efficient extraction, incremental updating, and personalization (Chhikara et al., 2025; Fang et al., 2025; Zhong et al., 2024). Weak abstraction and relational reasoning can hinder multi-hop and cross-context inference.

Explicitly structured memory systems organize memory into linked notes or graphs to support richer reasoning. A-MEM and O-Mem maintain evolving networks of notes via reflection and linking (Xu et al., 2025b; Wang et al., 2025; Park et al., 2023); MemGAS further introduces multi-granularity memory with GMM-based organization and entropy routing (Xu et al., 2025a). MIRIX partitions memory into specialized types managed by multiple agents (Wang & Chen, 2025), while Zep builds a temporally aware knowledge graph for cross-session consistency (Rasmussen et al., 2025). FluxMem selects among linear, graph, and hierarchical organizations using a reward-supervised MLP over conversation features. MemGAS instead routes across memory granularities using the entropy of query-relevance distributions; its GMM associates memories, whereas the BMM in FluxMem gates memory fusion.

3 METHODOLOGY

In this section, we introduce our framework, FLUXMEM, as illustrated in Fig. 2. It comprises a three-layer memory hierarchy to support long-horizon interactions and a BMM-based gating mechanism controls memory fusion and updates. In mid-term memory, multiple complementary structures are supported, with a conversation-aware selector that chooses the most suitable structure based on context. In the following parts, we first define the problem setting, then describe the three-layer memory hierarchy, multi-structure memory organization and retrieval, conversation-aware memory structure selection, and BMM-based memory fusion.

3.1 FORMALIZATION

We consider a long-horizon conversational setting where an agent interacts with a user over multiple turns. Let $H_t = \{p_1, \ldots, p_t\}$ denote the dialogue history up to turn t, where each *page* $p_i = (u_i, a_i)$

corresponds to a single user–agent exchange. Given query q_t, the agent generates response y_t using recent context and long-term conversational memories.

Memory Granularity and Hierarchy. We organize memory using a three-layer hierarchy operating at different temporal and semantic scales. *Short-term interaction memory* (STIM), denoted as $\mathcal{M}_t^S$, buffers the most recent pages to support fast recency-aware access. *Mid-term episodic memory* (MTEM), denoted as $\mathcal{M}_t^E = \{e_1, \ldots, e_N\}$, stores a set of episodic memory units, where each unit

$$e_j = \{p_{j,1}, \ldots, p_{j,m}\}, \tag{1}$$

corresponds to an episodic session composed of multiple semantically or temporally related pages. *Long-term semantic memory* (LTSM), denoted as $\mathcal{M}^L$, retains consolidated interaction knowledge.

Memory Write and Consolidation Pipeline. Newly generated pages are first written into short-term memory and then progressively organized and consolidated across memory layers. Formally, the memory update process can be abstracted as a stage-wise mapping:

$$p_i \xrightarrow{\phi_S} \mathcal{M}_t^S \;\rightarrow\; \mathcal{M}_t^E \xrightarrow{\phi_L} \mathcal{M}^L, \tag{2}$$

where ϕ_S and ϕ_L denote abstract write and consolidation operators. This pipeline balances fast access to recent interactions, structured episodic storage, and long-term semantic retention.

Structured Episodic Memory. Each episodic memory unit in MTEM is assigned a single suitable structure (linear, graph, or hierarchical) based on its conversational content, which determines how the unit is organized and retrieved. The selection of structure is treated as a context-dependent decision, allowing different episodic units to be organized under different structural forms.

Query and Response Generation. Given a query q_t, the agent retrieves and fuses relevant information across memory layers to form a memory-conditioned context:

$$\tilde{\mathcal{M}}_t = \Psi(q_t, \mathcal{M}_t^S, \mathcal{M}_t^E, \mathcal{M}^L), \tag{3}$$

where $\Psi(\cdot)$ denotes a memory retrieval and fusion function. The response is generated by conditioning on both the query and the retrieved memories:

$$y_t \sim p(y \mid q_t, \tilde{\mathcal{M}}_t). \tag{4}$$

3.2 Three-Layer Memory Hierarchy

Our three-level memory hierarchy comprises STIM, MTEM, and LTSM.

Short-Term Interaction Memory. STIM buffers the most recent dialogue history to provide fast access to recent context, which is usually most relevant to the user's next query. During retrieval, all STIM contents are directly incorporated into the generation context without filtering. However, an unbounded short-term buffer introduces noise and destabilizes downstream memory organization, necessitating capacity control. The design of STIM is motivated by findings from cognitive psychology (Cowan, 2001; 2010; Vogel & Machizawa, 2004; Mathy & Feldman, 2012), which show that human working memory can maintain only a few unstructured items, typically three to five chunks. Accordingly, we set the capacity of STIM to four pages to enforce a strict short-term memory budget. When this limit is exceeded, older pages are irreversibly transferred to MTEM for structured storage using a recency-based Least Recently Used (LRU) policy, reflecting the dominant role of recency in short-term conversational relevance. Formally, let $\mathcal{M}_t$ denote the set of interaction pages stored in STIM at turn t, and let C be the fixed STIM capacity. When $|\mathcal{M}_t| > C$, we evict a subset of pages

$$\mathcal{E}_t = \arg \min_{\mathcal{P} \subset \mathcal{M}_t,\, |\mathcal{P}| = |\mathcal{M}_t| - C} \sum_{p \in \mathcal{P}} \tau(p), \tag{5}$$

where $\tau(p)$ denotes the last access time of page p. The selected pages $\mathcal{E}_t$ are flushed to MTEM.

Importantly, transferring information to MTEM introduces a second, qualitatively different challenge. Unlike STIM, MTEM organizes interaction history into episodic units, where indiscriminately

appending every page would lead to fragmented or redundant structures. We therefore apply a probabilistic integration decision to determine whether a transferred page should be merged into an existing session and, if so, where it fits. This decision is implemented via a BMM, which guides integration using the distribution of similarity scores and applies a threshold to the resulting high-compatibility posterior. Details are provided in the Beta-Mixture-Gated Memory Fusion section.

Mid-Term Episodic Memory. MTEM serves as the primary repository for structured interaction history and bridges short-term context with long-term semantic memory. A key challenge is controlling the granularity and growth of stored information: storing individual interaction pages in isolation leads to fragmented memory representations and noisy retrieval, while indiscriminate accumulation results in unbounded memory expansion.

To address this challenge, MTEM organizes interaction history into episodic units, each grouping semantically related interaction pages into a coherent unit that serves as the basic unit for memory organization and retrieval. Each session contains (i) a sequence of pages with dialogue content, timestamps, representations, and continuity links; (ii) session-level metadata, including a summary and its representations; and (iii) indexing structures (e.g., entity–relation graphs or topic hierarchies) corresponding to the assigned memory structure. As MTEM grows, only a subset of units remains consistently useful. We therefore estimate the long-term utility of each session using a lightweight episodic utility score, $U(s) = w_1 c(s) + w_2 \ell(s) + w_3 d(s)$, which combines access frequency, interaction intensity, and recency. Prioritizing high-utility units for LTSM keeps MTEM compact.

Long-Term Semantic Memory. While MTEM preserves rich episodic interaction history, repeatedly retrieving episodic units is inefficient for information that remains stable across interactions. LTSM therefore consolidates high-utility episodic experiences into durable semantic knowledge for long-term reuse. Unlike episodic memory, LTSM stores abstracted and persistent information that is largely invariant to interaction context, such as user profiles, user-specific facts, general knowledge, and reusable interaction strategies. To maintain long-term memory under capacity constraints, LTSM employs an eligibility-based pruning mechanism rather than heuristic scalar scoring:

$$m \in \mathcal{M}_{\text{LTSM}} \;\text{ iff }\; u(m) \geq \tau_u \;\wedge\; r(m) \geq \tau_r \\ (\wedge\, c(m) \geq \tau_c), \tag{6}$$

where $u(m)$, $r(m)$, and $c(m)$ denote usage, recency, and optional confidence, and τ_* are the corresponding thresholds. Items failing these criteria are removed when capacity is exceeded, ensuring compact and reliable long-term storage. Together, STIM, MTEM, and LTSM dynamically organize transient context, episodic history, and stable semantic knowledge.

3.3 Multi-Structure Organization and Retrieval

Episodic units in MTEM are heterogeneous and best organized under different memory structures. Some units benefit from temporal organization, while others are better captured by relational or hierarchical structures. Therefore, imposing a single memory organization leads to retrieval errors across interaction scenarios. To address this, our framework maintains multiple complementary memory structures (Lewis et al., 2020; Collins & Quillian, 1969). *From a structural perspective*, we restrict our design space to linear, graph-based, and hierarchical memory structures as they constitute the foundational primitives of information organization. Classical data structure theory and information science show that complex information systems are fundamentally organized through sequential, hierarchical, or networked forms, while other structures can be reduced to or composed from these organizations (Cormen et al., 2022; Hopcroft et al., 1983; Simon, 2012; Buckland, 1991).

Linear Memory. Linear memory organizes episodic content chronologically, preserving the order of information acquisition. Semantic similarity and implicit recency guide retrieval for temporal queries, including step-by-step instructions and evolving goals.

Graph Memory. Graph memory organizes episodic content around relational structure (Edge et al., 2024). Retrieval is entity-centric, combining neighborhood expansion with semantic matching, and favors queries where relevance is defined by relational proximity rather than time.

Hierarchical Memory. Hierarchical memory organizes episodic content into multi-level abstractions (Sarthi et al., 2024). Retrieval follows a coarse-to-fine strategy by matching higher-level topics and accessing finer-grained content, supporting abstraction-aware recall across granularity levels.

Each memory structure emphasizes a different aspect of conversational information, including temporal, relational, and hierarchical aspects, motivating explicit structure selection. The next section describes how the agent dynamically selects among these structures based on context.

3.4 Context-Aware Memory Structure Selection

We formulate structure selection as a context-aware classification. At each turn t, the agent selects a structure matching the context and retrieval needs. Let $\mathcal{S} = \{\text{LINEAR}, \text{GRAPH}, \text{HIERARCHICAL}\}$ denote the supported structures. A lightweight selector chooses the structure.

Conversation feature representation. We represent the conversation state using a compact set of interpretable interaction-level features, rather than dense embeddings or heuristic rules. At each turn t, the context is encoded as $x_t \in \mathbb{R}^d$ capturing structural cues relevant to retrieval, including interaction scale, temporal signals, entity–relation patterns, and topic structure. This compact representation is processed by a small MLP for structure selection.

Structure selector. The selector is a shallow multi-layer perceptron (MLP) classifier that outputs a score for each candidate structure,

$$f_\theta(x_t) = \mathrm{Softmax}\big(g_\theta(x_t)\big), \tag{7}$$

where $g_\theta(\cdot)$ is the lightweight two-layer MLP. The predicted structure is

$$\hat{s}_t = \arg\max_{s \in \mathcal{S}} f_\theta(x_t)[s]. \tag{8}$$

The selected $\hat{s}_t$ is then used to organize episodic units in MTEM and guide subsequent retrieval.

Offline supervision via interaction-derived rewards. Since ground-truth labels for optimal memory structures are not directly available, we construct supervision signals offline using interaction-derived rewards. For each training interaction, all candidate structures are evaluated by running the agent under structure s and computing a scalar reward

$$r_t(s) = \lambda_q \, r_t^{\text{judge}}(s) + \lambda_m \, r_t^{\text{mem}}(s), \tag{9}$$

where r_t^{judge} denotes a response quality score and r_t^{mem} measures memory utilization effectiveness. The reward-maximizing structure, $s_t^* = \arg\max_{s \in \mathcal{S}} r_t(s)$, provides the target label.

The selector is trained offline by minimizing a standard supervised classification loss,

$$\mathcal{L}_{\text{sel}}(\theta) = \sum_t \ell\big(f_\theta(x_t),\, s_t^*\big), \tag{10}$$

where $\ell(\cdot, \cdot)$ denotes the cross-entropy loss.

At inference stage, the trained selector predicts the most suitable memory structure for the current interaction context. As the conversation evolves, structure selection is performed repeatedly, allowing memory organization to adapt dynamically to changing interaction patterns.

3.5 Beta-Mixture-Gated Memory Fusion

In long-horizon interactions, deciding whether newly arriving information should be merged into existing memory is unstable. Fixed similarity thresholds or heuristic top-K rules are brittle, as similarity score distributions vary across conversation types and interaction stages. A *BMM-based gate* provides a soft, distribution-aware fusion criterion.

Given an incoming interaction summary and a set of candidate episodic units, we compute a matching score for each candidate and normalize it to $x \in (0, 1)$. These normalized scores are modeled using a two-component Beta mixture (Arazo et al., 2019):

$$p(x) = \pi \, \mathrm{Beta}(x; \alpha_1, \beta_1) + (1 - \pi) \, \mathrm{Beta}(x; \alpha_0, \beta_0), \tag{11}$$

where two components correspond to high- and low-compatibility regimes. Here, π denotes the mixture weight, and (α_k, β_k) parameterize the Beta distribution of component k. Mixture parameters are estimated via an EM procedure with responsibility-weighted moment updates (Dempster et al., 1977). Based on the mixture, we compute the posterior probability,

$$\begin{aligned} g(x) &= P(z = 1 \mid x) \\ &= \frac{\pi \, \mathrm{Beta}(x; \alpha_1, \beta_1)}{\sum_{k \in \{0,1\}} \pi_k \, \mathrm{Beta}(x; \alpha_k, \beta_k)}, \end{aligned} \tag{12}$$

where $z \in \{0, 1\}$ is a latent component indicator and $z = 1$ denotes the high-compatibility component. The posterior $g(x)$ serves as a soft gating signal that measures how likely a candidate belongs to the high-compatibility regime under the learned score distribution.

Importantly, fusion decisions are made in the posterior probability space rather than on raw similarity scores. Candidates with high posterior mass under the high-compatibility component are retained, with a minimum-keep safeguard to avoid over-filtering. Retained candidates are merged into the most compatible episodic unit under the selected memory structure, or stored as a new unit if none qualify. The posterior threshold τ_{BMM} controls retention, while the fitted mixture adapts the mapping from matching scores to posterior probabilities. The gate therefore remains threshold-dependent; its selectivity is examined in the parameter sensitivity analysis.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets

To evaluate the effectiveness of our proposed framework, we conduct a series of experiments across two representative datasets: LoCoMo (Maharana et al., 2024) and PERSONAMEM (Jiang et al., 2025a). Both datasets test long-horizon reasoning and memory in LLM agents.

4.1.2 Baselines

We compare *FluxMem* against a range of state-of-the-art memory systems, including: **LangMem** (LangChain, 2025): A language model with built-in memory capabilities. **Mem0** (Chhikara et al., 2025): A flat retrieval-based memory system. **ZEP** (Rasmussen et al., 2025): A time-aware graph memory system. **MemoryOS** (Kang et al., 2025): An OS-inspired memory management system. **A-Mem** (Xu et al., 2025b): An explicit structured memory system using linked notes. **O-mem** (Wang et al., 2025): A graph-based memory organization method. **MemR^3** (Du et al., 2025): A closed-loop controller that alternates between retrieval, reflection, and answering to optimize evidence coverage for LLM agents. **MEMOS** (Li et al., 2025b): A redundancy-filtering memory system for edge settings. **HippoRAG 2** (Gutiérrez et al., 2025): Graph-based retrieval.

4.1.3 Implementation Details

Experiments are conducted on 2×A100 GPUs using GPT-4.1 with temperature 0. We employ a three-layer memory: STIM (capacity 4, LRU), MTEM (up to 2000 episodic sessions with utility-aware pruning), and LTSM (up to 100 consolidated entries retrieved by semantic similarity). Memory fusion in MTEM is governed by a two-component Beta Mixture Model gate, where matching scores are filtered by the high-compatibility posterior. The saved LoCoMo runs use a posterior threshold of 0.5 and minimum retention of 1. The memory structure selector is a two-layer MLP (12-dim input, hidden size 4) with a 3-way output over LINEAR/GRAPH/HIERARCHICAL, trained with interaction-derived supervision. Reciprocal rank fusion combines all-MiniLM-L6-v2 with BM25.

4.1.4 Evaluation Metrics

We evaluate using dataset-specific metrics. On PERSONAMEM, where each query is a single-choice question, we use accuracy. On LOCOMO, which involves free-form response generation, we assess responses using F1, BLEU-1 (B1), ROUGE-L, ROUGE-1, ROUGE-2, and BERTScore.

Table 1: PERSONAMEM accuracy on six reported query types. Average is their unweighted mean; latest preferences are excluded. Best in **bold**, second-best underlined.

Method	Recall user shared facts	Suggest new ideas	Track full preference evolution	Revisit reasons behind preference updates	Provide preference-aligned recommendations	Generalize to new scenarios	**Average**
LangMem	31.29	24.73	53.24	81.82	40.00	8.77	39.98
Mem0	32.13	15.05	54.68	80.81	52.73	57.89	48.88
A-Mem	63.01	27.96	54.68	85.86	69.09	57.89	59.75
Memory OS	72.72	17.20	58.27	78.79	**72.72**	56.14	59.31
O-Mem	67.81	21.51	61.15	89.90	65.45	73.68	63.25
HippoRAG 2	76.74	33.33	56.12	78.79	23.64	38.60	51.20
FluxMem	**85.27**	**36.56**	**65.47**	**91.92**	70.91	**82.46**	**72.10**
Improvement (pp)	+8.53	+3.23	+4.32	+2.02	-1.81	+8.78	+8.85

Table 2: Performance comparison on LoCoMo with GPT-4.1. Average is the unweighted mean of the four categories. Best in **bold**, second-best underlined. Improvements are absolute score differences from the strongest baseline in each column.

Method	Multi-hop			Temporal			Open			Single-hop			Average		
	F1	B-1	R-L	F1	B-1	R-L	F1	B-1	R-L	F1	B-1	R-L	F1	B-1	R-L
LangMem	41.11	32.09	37.71	53.67	46.22	47.37	33.38	27.26	24.55	51.13	44.22	48.37	44.82	37.45	39.50
Mem0	28.84	20.22	27.45	24.21	18.75	23.23	23.59	19.09	23.13	32.74	27.58	33.02	27.35	21.41	26.71
ZEP	8.74	5.98	6.79	7.40	4.70	6.07	7.64	3.46	5.18	7.91	3.80	6.26	7.92	4.49	6.08
MemoryOS	43.03	34.09	39.75	44.43	32.03	42.64	28.36	23.24	27.83	49.31	42.61	49.84	41.28	32.99	40.02
A-Mem	32.41	24.28	31.26	37.23	33.02	39.61	16.16	14.64	15.19	40.94	35.54	42.32	31.69	26.87	32.10
MEMOS	20.27	13.35	15.75	14.86	7.86	11.58	11.87	6.85	9.32	18.96	11.42	17.37	16.49	9.87	13.51
MemR^3	18.65	11.17	14.37	25.39	15.90	22.63	11.56	5.77	8.96	24.35	12.72	22.00	19.99	11.39	16.99
O-mem	42.64	34.08	32.78	**57.48**	**49.76**	48.88	30.58	25.69	21.11	54.89	48.98	44.26	46.40	39.63	36.76
HippoRAG 2	28.11	18.92	27.68	10.35	6.92	9.82	24.91	21.10	24.48	44.82	37.25	44.37	27.05	21.05	26.59
FluxMem	**48.56**	**39.90**	**44.05**	56.67	41.88	**54.89**	**37.30**	**29.63**	**36.76**	**62.12**	**53.52**	**62.33**	**51.16**	**41.23**	**49.51**
Improvement (pp)	+5.53	+5.81	+4.30	-0.81	-7.88	+6.01	+3.92	+2.37	+8.93	+7.23	+4.54	+12.49	+4.76	+1.60	+9.49

4.2 Overall Performance

Tables 1 and 2 report the overall results on PersonaMem and LoCoMo. Overall, **FluxMem** achieves the strongest and most consistent performance across both datasets, demonstrating its effectiveness for long-horizon conversational memory. On PersonaMem (Table 1), FluxMem attains the highest average accuracy (72.10%), exceeding the strongest baseline average by 8.85 percentage points, with clear advantages on preference-centric tasks such as tracking preference evolution, revisiting update reasons, and generalizing to new scenarios. On LoCoMo (Table 2), FluxMem again delivers the best overall results, achieving the highest average F1, BLEU-1, and ROUGE-L scores, with strong gains in multi-hop, open, and single-hop reasoning. In contrast, competing methods relying on fixed memory structures or layer-specific heuristics perform well only in isolated settings and struggle to adapt to varying interaction patterns and retrieval requirements. In a few categories, FluxMem does not outperform the strongest baselines, which is expected as certain tasks align closely with specific fixed structures. This reflects a trade-off between specializing for

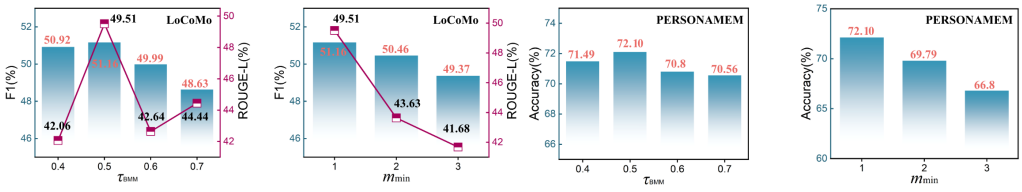

Figure 3: Parameter experiments results.

Table 3: Ablation study on PERSONAMEM.

Method	Recall user shared facts	Suggest new ideas	Track full preference evolution	Revisit reasons behind preference updates	Provide preference-aligned recommendations	Generalize to new scenarios
w/o Linear	80.62↓4.7	**40.86**↑4.3	62.59↓2.9	90.91↓1.0	69.09↓1.8	75.44↓7.0
w/o Graph	78.29↓7.0	34.41↓2.2	63.31↓2.2	86.87↓5.1	67.27↓3.6	75.44↓7.0
w/o Hierarchy	79.84↓5.4	35.48↓1.1	61.87↓3.6	90.91↓1.0	63.64↓7.3	78.95↓3.5
w/o BMM	81.40↓3.9	37.63↑1.1	61.87↓3.6	87.88↓4.0	69.09↓1.8	78.95↓3.5
FluxMem	**85.27**	36.56	**65.47**	**91.92**	**70.91**	**82.46**

Table 4: Ablation study on LoCoMo with GPT-4.1.

Method	Multi-hop			Temporal			Open			Single-hop		
	F1	B-1	R-L	F1	B-1	R-L	F1	B-1	R-L	F1	B-1	R-L
w/o Linear	44.39↓4.2	36.63↓3.3	36.73↓7.3	49.86↓6.8	36.76↓5.1	49.53↓5.4	29.60↓7.7	6.67↓23	27.41↓9.4	58.75↓3.4	47.26↓6.3	58.49↓3.8
w/o Graph	41.95↓6.6	32.23↓7.7	32.31↓12	53.95↓2.7	41.38↓0.5	54.12↓0.8	17.86↓19	4.24↓25	17.76↓19	61.39↓0.7	52.65↓0.9	61.63↓0.7
w/o Hierarchy	42.36↓6.2	32.00↓7.9	35.38↓8.7	55.50↓1.2	**43.97**↑2.1	**55.28**↑0.4	25.28↓12	4.89↓25	25.64↓11	58.39↓3.7	47.89↓5.6	59.23↓3.1
w/o BMM	41.18↓7.4	31.70↓8.2	32.48↓12	54.33↓2.3	41.75↓0.1	54.61↓0.3	20.97↓16	3.79↓26	20.73↓16	59.74↓2.4	47.74↓5.8	59.05↓3.3
FluxMem	**48.56**	**39.90**	**44.05**	**56.67**	41.88	54.89	**37.30**	**29.63**	**36.76**	**62.12**	**53.52**	**62.33**

individual scenarios and maintaining robust performance across diverse long-horizon interaction patterns. These results show that multi-structure episodic memory with context-adaptive structure selection yields more stable and robust improvements than fixed-structure systems.

4.3 Ablation Studies

We conduct ablation studies on PERSONAMEM and LOCOMO to examine individual components (Tables 3 and 4). Removing linear structure causes the largest drop on temporally dependent tasks, highlighting the importance of chronological organization. Eliminating graph or hierarchical structure mainly affects relational and abstraction-oriented tasks, respectively. Removing BMM-based gating also degrades performance, indicating the importance of distribution-aware fusion. On LOCOMO, all ablated variants underperform the full model, with clear drops in multi-hop and open reasoning. The pronounced degradation on OPEN is expected, as open-domain questions combine weak temporal cues with sparse relational signals, making them dependent on flexible structure selection. Notably, *w/o BMM* consistently lags behind FluxMem, underscoring the benefit of adaptive gating over heuristic thresholds. FluxMem combines diverse memory structures, adaptive selection, and BMM gating to support robust long-horizon reasoning.

4.4 Analysis of Parameter Sensitivity

We analyze the BMM posterior threshold τ_{BMM} and minimum retention $m_{\min}$ (Fig. 3). (1) τ_{BMM}: Fixing $m_{\min} = 1$, we vary τ_{BMM} from 0.4 to 0.7. Performance peaks at $\tau_{\text{BMM}} = 0.5$ on both datasets; smaller thresholds admit noise, while larger ones become overly restrictive. (2) $m_{\min}$: Fixing $\tau_{\text{BMM}} = 0.5$, we vary $m_{\min}$ from 1 to 3. Larger $m_{\min}$ retains redundant memories and degrades performance. FluxMem favors $\tau_{\text{BMM}} = 0.5$ and $m_{\min} = 1$.

5 Conclusion

We introduce FluxMem, a multi-structure conversational memory framework with adaptive structure selection for long-horizon interactions. By integrating linear, graph, and hierarchical memories with conversation-aware selection and BMM-based fusion, FluxMem supports diverse reasoning demands. Experiments on PERSONAMEM and LOCOMO demonstrate consistent improvements over baselines, highlighting the effectiveness of adaptive memory structuring for long-horizon agents. Our analysis indicates that no single memory structure is optimal across interaction patterns,and that choosing the structure from conversation-level context is more effective than committing to a fixed organization. Future work includes adapting the selector online as interactions unfold and enlarging the space of candidate structures.

AI USE STATEMENT

Generative AI tools were used primarily for language polishing. The authors take responsibility for the final content of the paper.

ETHICS STATEMENT

This work studies conversational memory using the PersonaMem and LoCoMo benchmarks. Long-term memory systems can retain sensitive personal information and propagate outdated or inaccurate user profiles. Benchmark performance does not by itself establish suitability for deployment in sensitive settings.

REPRODUCIBILITY STATEMENT

Section 3 describes the FluxMem framework, and Section 4 reports the experimental setup. Appendix B describes the datasets, baselines, and evaluation metrics. Additional details cover BMM gating (Appendix C), conversation features (Appendix D), selector training (Appendix E), memory structure implementations (Appendix G), and LLM prompts (Appendix O). These materials specify the procedures and configurations underlying the reported experiments.

REFERENCES

Susmit Agrawal, Krishn Vishwas Kher, Saksham Mittal, Swarnim Maheshwari, and Vineeth N Balasubramanian. Memory-integrated reconfigurable adapters: A unified framework for settings with multiple tasks. *arXiv preprint arXiv:2512.00940*, 2025.

Anonymous. REMem: Reasoning with episodic memory in language agent, 2026. Submitted to ICLR 2026.

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 312–321, 2019.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations (ICLR)*, 2024.

Michael K Buckland. Information as thing. *Journal of the American Society for information science*, 42(5):351–360, 1991.

Dillan Cellier, Isaac T Petersen, and Kai Hwang. Dynamics of hierarchical task representations. *Journal of Neuroscience*, 42(38):7276–7284, 2022.

Junyi Chen, Mengjia Wu, Qian Liu, and Yi Zhang. Explainable prediction of knowledge recombination: A synergized method with heterogeneous hypergraph learning and large language models. *Information Processing & Management*, 63(1):104336, 2026. ISSN 0306-4573. doi: https://doi.org/10.1016/j.ipm.2025.104336. URL https://www.sciencedirect.com/science/article/pii/S0306457325002778.

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

Allan M Collins and M Ross Quillian. Retrieval time from semantic memory. *Journal of verbal learning and verbal behavior*, 8(2):240–247, 1969.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

Nelson Cowan. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and brain sciences*, 24(1):87–114, 2001.

Nelson Cowan. The magical mystery four: How is working memory capacity limited, and why? *Current directions in psychological science*, 19(1):51–57, 2010.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1): 1–38, 1977.

Xingbo Du, Loka Li, Duzhen Zhang, and Le Song. Memr^3: Memory retrieval via reflective reasoning for llm agents. *arXiv preprint arXiv:2512.20237*, 2025.

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

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

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: Non-parametric continual learning for large language models. In *Forty-second International Conference on Machine Learning*, 2025.

John E Hopcroft, Jeffrey D Ullman, and Alfred Vaino Aho. *Data structures and algorithms*, volume 175. Addison-wesley Boston, MA, USA:, 1983.

Runsong Jia, Mengjia Wu, Ying Ding, Jie Lu, and Yi Zhang. HetGCoT: Heterogeneous graph-enhanced chain-of-thought LLM reasoning for academic question answering. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 15950–15963, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.864. URL https://aclanthology.org/2025.findings-emnlp.864/.

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

Jiantong Jiang, Peiyu Yang, Rui Zhang, and Feng Liu. Towards efficient large language model serving: A survey on system-aware kv cache optimization. *Authorea Preprints*, 2025b.

Aneesh Jonelagadda, Christina Hahn, Haoze Zheng, and Salvatore Penachio. Mnemosyne: An unsupervised, human-inspired long-term memory architecture for edge-based llms. *arXiv preprint arXiv:2510.08601*, 2025.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory os of ai agent. *arXiv preprint arXiv:2506.06326*, 2025.

LangChain. langmem: Language model memory. https://github.com/langchain-ai/langmem, 2025.

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

Junnan Li, Richard Socher, and Steven C. H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Xin Li, Zhuo Cai, Shoujin Wang, Kun Yu, and Fang Chen. A survey on enhancing causal reasoning ability of large language models. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 399–416. Springer, 2025a.

Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, et al. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459*, 2024.

Zhiyu Li, Chenyang Xi, Chunyu Li, Ding Chen, Boyu Chen, Shichao Song, Simin Niu, Hanyu Wang, Jiawei Yang, Chen Tang, et al. Memos: A memory os for ai system. *arXiv preprint arXiv:2507.03724*, 2025b.

Lei Liu, Xiaoyan Yang, Yue Shen, Binbin Hu, Zhiqiang Zhang, Jinjie Gu, and Guannan Zhang. Think-in-memory: Recalling and post-thinking enable llms with long-term memory. *arXiv preprint arXiv:2311.08719*, 2023.

Mingfei Lu, Mengjia Wu, Jiawei Xu, Weikai Li, Feng Liu, Ying Ding, Yizhou Sun, Jie Lu, and Yi Zhang. From newborn to impact: Bias-aware citation prediction, 2025. URL https://arxiv.org/abs/2510.19246.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024.

Fabien Mathy and Jacob Feldman. What's magic about magic numbers? chunking and data compression in short-term memory. *Cognition*, 122(3):346–362, 2012.

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

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

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2023.

Hongjin Qian, Zheng Liu, Kelong Mao, Yujia Zhou, and Zhicheng Dou. Memorag: Moving towards next-gen rag via memory-inspired knowledge discovery. In *Proceedings of the ACM Web Conference (WWW)*, 2025.

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

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

Katja Schlegel, Nils R Sommer, and Marcello Mortillaro. Large language models are proficient in solving and creating emotional intelligence tests. *Communications Psychology*, 3(1):80, 2025.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Herbert A Simon. The architecture of complexity. In *The Roots of Logistics*, pp. 335–361. Springer, 2012.

Juliana E Trach and Samuel D McDougle. Mental graphs structure the storage and retrieval of visuomotor associations. *Nature Human Behaviour*, pp. 1–15, 2025.

Edward K Vogel and Maro G Machizawa. Neural activity predicts individual differences in visual working memory capacity. *Nature*, 428(6984):748–751, 2004.

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

Piaohong Wang, Motong Tian, Jiaxian Li, Yuan Liang, Yuqing Wang, Qianben Chen, Tiannan Wang, Zhicong Lu, Jiawei Ma, Yuchen Eleanor Jiang, et al. O-mem: Omni memory system for personalized, long horizon, self-evolving agents. *arXiv e-prints*, pp. arXiv–2511, 2025.

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

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations (ICLR)*, 2025a.

MJ Wu, Yi Zhang, Robin Haunschild, and Lutz Bornmann. Leveraging large language models for post-publication peer review: Potential and limitations. In *Proceedings of the 20th International Conference on Scientometrics & Informetrics (ISSI 2025)*, pp. 1207–1226, 2025b.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, wenlin zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, and Tong Xu. From single to multi-granularity: Toward long-term memory association and selection of conversational agents, 2025a. URL https://arxiv.org/abs/2505.19549.

Jing Xu, Arthur Szlam, and Jason Weston. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 5180–5197, 2022.

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

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

Xiaoyu Yang, Jie Lu, and En Yu. Adapting multi-modal large language model to concept drift from pre-training onwards. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=b20VK2GnSs.

Weizhi Zhang, Xinyang Zhang, Chenwei Zhang, Liangwei Yang, Jingbo Shang, Zhepei Wei, Henry Peng Zou, Zijie Huang, Zhengyang Wang, Yifan Gao, Xiaoman Pan, Lian Xiong, Jingguo Liu, Philip S. Yu, and Xian Li. Personaagent: When large language model agents meet personalization at test time, 2025a. URL https://arxiv.org/abs/2506.06254.

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. *ACM Transactions on Information Systems*, 43(6):1–47, 2025b.

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

APPENDIX CONTENTS

A Related Work **15**

B Description of Datasets, Baselines and Evaluation Metrics **15**

B.1 Datasets 15

B.2 Baselines 16

B.3 Evaluation Metrics 17

C Beta-Mixture Gating Model (BMM) **17**

D Conversation Feature Definition for Structure Selection **19**

D.1 Feature Overview 19

D.2 Feature Definition Table 19

E Structure Selector Training Pipeline **20**

F Additional Experimental Results **21**

G Memory Structure Implementations **21**

H Computational Cost Analysis **22**

I Homogeneous Ensemble Comparison **23**

J Cross-Backbone Evaluation **24**

K Robustness and Variance Analysis **24**

L Reward Weighting Sensitivity **24**

M Structure Selector Analysis **25**

N Case Studies **26**

O LLM Prompts Design **27**

A Related Work

LLMs have been widely applied across diverse domains, including dialogue systems, decision-making agents, code generation, and scientific reasoning(Yang et al., 2025; Li et al., 2025a; Jia et al., 2025; Chen et al., 2026; Wu et al., 2025b; Lu et al., 2025; Jiang et al., 2025b). Their strong in-context learning and generative capabilities make them a natural foundation for autonomous agent systems that require long-horizon interaction and memory management. Recent research on long-term memory for LLM agents falls into three paradigms: (1) flat retrieval-based systems, (2) explicitly structured systems, and (3) policy-managed memory systems.

Flat retrieval-based memory systems store facts, events, or summaries in flat repositories for similarity-based recall. Mem0, LightMem, and MemoryBank emphasize efficient extraction, incremental updating, and personalization (Chhikara et al., 2025; Fang et al., 2025; Zhong et al., 2024). Limited abstraction and relational reasoning can hinder multi-hop and cross-context inference.

Explicitly structured memory systems organize memory into linked notes or graphs to support richer reasoning. A-MEM and O-Mem maintain evolving networks of notes via reflection and linking (Xu et al., 2025b; Wang et al., 2025); MemGAS further introduces multi-granularity memory with GMM-based organization and entropy routing (Xu et al., 2025a). MIRIX partitions memory into specialized types managed by multiple agents (Wang & Chen, 2025), while Zep builds a temporally aware knowledge graph for cross-session consistency (Rasmussen et al., 2025). FluxMem selects among linear, graph, and hierarchical organizations using a reward-supervised MLP over conversation features. MemGAS uses query-relevance entropy to route across granularities; its GMM associates memories, while FluxMem uses BMM fusion gating.

Policy managed memory systems optimize memory behavior via learned policies or OS-inspired mechanisms. RL-based Memory-R1 and MIRA learn what to write/keep based on long-horizon rewards (Yan et al., 2025; Agrawal et al., 2025). MemoryOS adopts concepts like hierarchical storage and eviction (Kang et al., 2025), and Mnemosyne targets edge settings with redundancy pruning and probabilistic recall (Jonelagadda et al., 2025). Most approaches retain fixed structural biases or hand-crafted rules, motivating learnable structure decisions.

B Description of Datasets, Baselines and Evaluation Metrics

B.1 Datasets

LoCoMo (Maharana et al., 2024) is a benchmark designed to evaluate LLM agents' long-term conversational memory. It consists of very long dialogue traces, where each conversation contains on average $\sim$300 turns and $\sim$9,000 tokens, spanning months of interactions (typically 19–35 sessions). LoCoMo groups memory and reasoning questions into five categories:

- **Single-hop reasoning:** the answer can be retrieved from information in a single session.
- **Multi-hop reasoning:** requires combining evidence across sessions.
- **Temporal reasoning:** tests temporal cues and time-gap reasoning in dialogue.
- **Open-domain:** combines user facts with external knowledge or commonsense.
- **Adversarial:** tests recognition of unanswerable questions without hallucination.

PERSONAMEM (Jiang et al., 2025a) is a benchmark designed to evaluate LLMs' ability to perform *dynamic user profiling* and generate personalized responses over long-horizon interactions. It consists of multi-session dialogue histories in which user attributes and preferences evolve over time due to life events or contextual changes. We report six PERSONAMEM query types:

- **Recall user-shared facts:** tests recall of static facts explicitly shared by the user.
- **Track preference evolution:** tests reconstruction of preference changes across sessions.
- **Revisit reasons behind preference updates:** tests whether the model can recall the underlying events or causes that led to preference changes.
- **Suggest new ideas:** tests novel suggestions aligned with inferred user preferences.

- **Provide preference-aligned recommendations:** evaluates personalized recommendation quality conditioned on the current user profile.
- **Generalize to new scenarios:** tests whether user-specific knowledge learned in one domain can be transferred to unseen tasks or contexts.

The average in Table 1 is the unweighted mean over these six category accuracies.

B.2 BASELINES

Mem0 (Chhikara et al., 2025) is a scalable long-term memory architecture specifically designed for AI agents to address the challenge that large language models cannot maintain consistency in prolonged multi-turn conversations due to fixed context window limitations. This method comprises two core stages: "extraction" and "update". First, it leverages LLMs combined with context to extract key facts from new conversations. Subsequently, through intelligent decision-making (Tool Call), it executes ADD, UPDATE, DELETE, or NOOP instructions to dynamically maintain memory consistency. Additionally, Mem0 provides an enhanced variant ($Mem0^s$) that integrates graph databases by using entity nodes and relationship edges to capture complex information structures.

LangMem (LangChain, 2025) is a long-term memory management library developed by the LangChain team to help AI agents learn and adapt from interactions. It provides tools for extracting important information from conversations, optimizing agent behavior, and maintaining long-term memory, with support for arbitrary storage systems and native LangGraph integration. Features include tools for recording and searching memories, background knowledge extraction and integration, and LangGraph support for cross-session personalization.

MemoryOS (Kang et al., 2025) is an OS-inspired memory management framework that optimizes long-term memory for LLM agents through a multi-tiered architecture. It includes short-term, mid-term, and long-term memory layers, allowing dynamic updates and integration during conversations. The framework features memory storage, update, retrieval, and generation modules, facilitating efficient memory decay and redundancy filtering. By applying OS principles, MemoryOS enhances response consistency and coherence while minimizing computational overhead.

A-MEM (Xu et al., 2025b) is an agentic memory architecture that organizes memory as a network of interconnected notes. It employs dedicated agents to manage note creation, linking, and retrieval, enabling dynamic memory evolution through reflection and integration of new information. Structured links between notes support hierarchical reasoning, relational reasoning, and recall.

ZEP (Rasmussen et al., 2025) is a time-aware graph memory architecture designed for LLM agents to maintain consistency across multi-session interactions. It constructs a knowledge graph that captures temporal relationships between memory entries, allowing agents to reason about changes over time. ZEP employs specialized retrieval mechanisms to access relevant information based on temporal context, enhancing the agent's ability to provide coherent and contextually appropriate responses.

MEMOS (Li et al., 2025b) is a memory operating system designed to treat memory as a schedulable and evolvable system resource for Large Language Models (LLMs). It establishes a unified hierarchy that integrates three distinct memory types—plaintext, activation, and parameter memory—encapsulated within a standardized unit called the *MemCube*. By employing a layered architecture featuring a scheduler and lifecycle manager, MEMOS enables the dynamic transformation, storage, and retrieval of memory, thereby facilitating long-term consistency, adaptive personalization, and efficient knowledge evolution across complex tasks.

MemR^3 (Du et al., 2025) is an autonomous memory retrieval system designed to transform the standard retrieve-then-answer pipeline into a closed-loop sequential decision process for LLM agents. Implemented as a LangGraph workflow, it utilizes a router to dynamically select actions among 'retrieve', 'reflect', and 'answer' to optimize response quality. Central to its architecture is a global evidence-gap tracker that explicitly maintains verified evidence and identifies missing information gaps throughout the interaction. This design enables the system to iteratively refine queries and stop retrieval early once sufficient information is gathered, functioning as a plug-and-play controller compatible with various existing memory backends.

O-Mem (Wang et al., 2025) is an omni-memory framework designed to enhance the long-term interaction and dynamic personalization of LLM agents. Unlike traditional systems that passively

store interaction history, O-Mem utilizes an *active user profiling* mechanism that dynamically extracts and updates user characteristics and event records during ongoing dialogues. It employs a hierarchical retrieval strategy across three distinct components—Persona Memory (for long-term attributes), Working Memory (for topical context), and Episodic Memory (for clue-triggered recall)—to construct a holistic user model. This architecture maintains long-horizon consistency and updates its user model for adaptive, coherent personalization.

HippoRAG (Gutiérrez et al., 2025) is a neurobiologically inspired framework that repurposes Retrieval-Augmented Generation (RAG) as a non-parametric continual learning system for LLMs. Grounded in the hippocampal indexing theory, it maintains a dynamic Knowledge Graph (KG) that continuously integrates new information through simulated synaptic plasticity, rather than static vector indexing. By employing the Personalized PageRank (PPR) algorithm for retrieval, HippoRAG enables pattern completion and complex multi-hop reasoning across temporally distributed facts, supporting one-pass knowledge updates without parametric training or catastrophic forgetting.

B.3 Evaluation Metrics

F1: For the saved FluxMem LoCoMo predictions, the evaluator lowercases each answer, extracts word tokens with a regular expression, and computes overlap between sets of unique tokens. Scores are averaged over questions within each category; the main table reports an unweighted average over the four included categories. F1 is the harmonic mean of precision and recall:

$$F1 = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{13}$$

where Precision $= \frac{TP}{TP+FP}$ and Recall $= \frac{TP}{TP+FN}$, with TP, FP, and FN denoting true positives, false positives, and false negatives, respectively.

Bleu-1: BLEU (Bilingual Evaluation Understudy) is a metric for evaluating machine translation and text generation quality. BLEU-1 measures the proportion of unigrams (single words) in the generated text that appear in the reference text:

$$\text{BLEU-1} = BP \cdot \exp\left(\sum_{n=1}^{1} w_n \log p_n\right), \tag{14}$$

where

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{1-r/c} & \text{if } c \le r \end{cases}, \tag{15}$$

$$p_n = \frac{\sum_i \sum_k \min(h_{ik}, m_{ik})}{\sum_i \sum_k h_{ik}}, \tag{16}$$

Here, BP denotes the brevity penalty used to penalize short system outputs, where c is the length of the candidate translation and r is the effective reference length. The term w_n represents the weight assigned to n-gram precision p_n. Specifically, h_{ik} is the count of the k-th n-gram in the i-th candidate sentence, while m_{ik} is its maximum frequency in the corresponding reference. The $\min(h_{ik}, m_{ik})$ function clips counts to prevent repeated words from inflating precision.

C Beta-Mixture Gating Model (BMM)

Score normalization. Given an incoming memory item, we compute matching scores against n candidate sessions, producing $\{s_i\}_{i=1}^{n}$. For Beta modeling, min–max normalization maps scores to $(0, 1)$:

$$x_i = \begin{cases} \epsilon + (1 - 2\epsilon)\dfrac{s_i - s_{\min}}{s_{\max} - s_{\min}}, & s_{\max} > s_{\min}, \\ 0.5, & s_{\max} = s_{\min}, \end{cases} \qquad x_i \in (0, 1), \tag{17}$$

where $s_{\min} = \min_i s_i$, $s_{\max} = \max_i s_i$, and ϵ is a small constant to avoid boundary values.

Two-component Beta mixture. We fit a two-component Beta mixture model to $\{x_i\}$:

$$p(x) = \pi_0 \,\mathrm{Beta}(x; \alpha_0, \beta_0) + \pi_1 \,\mathrm{Beta}(x; \alpha_1, \beta_1), \qquad \pi_k \geq 0, \sum_{k\in\{0,1\}} \pi_k = 1, \tag{18}$$

with

$$\mathrm{Beta}(x; \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}. \tag{19}$$

We interpret the two components as *low-compatibility* vs. *high-compatibility* regimes.

Choice of mixture family. The Beta family matches the bounded support of normalized compatibility scores and can represent skewed score distributions. This modeling choice is distinct from DivideMix (Li et al., 2020), which uses a Gaussian mixture to separate per-sample training losses in noisy-label learning. Its mixture-based separation motivates a related probabilistic perspective, but does not establish that a Beta mixture is preferable for conversational memory fusion.

EM fitting with log-space responsibilities. We estimate parameters via EM for a fixed number of iterations (or until convergence). Introduce latent assignments $z_i \in \{0, 1\}$ and compute responsibilities $r_{ik} = p(z_i = k \mid x_i)$.

INITIALIZATION (QUANTILE-BASED). We initialize the two components to be separated in mean by using the empirical 30th and 70th percentiles of $\{x_i\}$. Let $q_{0.3}$ and $q_{0.7}$ denote these quantiles. We initialize component means as $\mu_0 \leftarrow q_{0.3}$ and $\mu_1 \leftarrow q_{0.7}$ and set initial concentrations (e.g., via a small fixed variance), then obtain (α_k, β_k) by moment matching.

E-STEP. For numerical stability, we compute responsibilities in log-space:

$$\log \tilde{r}_{ik} = \log \pi_k + \log \mathrm{Beta}(x_i; \alpha_k, \beta_k), \qquad r_{ik} = \frac{\exp(\log \tilde{r}_{ik})}{\sum_{k'\in\{0,1\}} \exp(\log \tilde{r}_{ik'})}. \tag{20}$$

Define effective counts $N_k = \sum_{i=1}^{n} r_{ik}$.

M-STEP (MIXTURE WEIGHTS).

$$\pi_k \leftarrow \frac{N_k}{n}. \tag{21}$$

M-STEP (BETA PARAMETERS VIA RESPONSIBILITY-WEIGHTED MOMENTS). Rather than numerically optimizing the Beta log-likelihood, we adopt a lightweight update using weighted moments. Compute the responsibility-weighted mean and variance:

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{n} r_{ik} x_i, \qquad \sigma_k^2 = \frac{1}{N_k} \sum_{i=1}^{n} r_{ik}(x_i - \mu_k)^2. \tag{22}$$

For $\mathrm{Beta}(\alpha, \beta)$, $\mu = \alpha/(\alpha+\beta)$ and $\sigma^2 = \frac{\mu(1-\mu)}{\kappa+1}$ with concentration $\kappa = \alpha + \beta$. Thus,

$$\kappa_k \leftarrow \frac{\mu_k(1-\mu_k)}{\sigma_k^2} - 1, \qquad \alpha_k \leftarrow \mu_k \kappa_k, \qquad \beta_k \leftarrow (1-\mu_k)\kappa_k. \tag{23}$$

In practice, we clamp σ_k^2 away from zero and clip $\kappa_k, \alpha_k, \beta_k$ to avoid degenerate updates.

Identifying the high-compatibility component. After fitting, we designate the component with larger Beta mean as the *high* component:

$$k^\star = \arg\max_{k\in\{0,1\}} \frac{\alpha_k}{\alpha_k + \beta_k}. \tag{24}$$

The gate value for a candidate score x is the posterior of this high component:

$$g(x) = p(z = k^\star \mid x) = \frac{\pi_{k^\star} \mathrm{Beta}(x; \alpha_{k^\star}, \beta_{k^\star})}{\sum_{k\in\{0,1\}} \pi_k \mathrm{Beta}(x; \alpha_k, \beta_k)}. \tag{25}$$

Algorithm 1 Beta-Mixture-Gated Memory Fusion

Require: incoming item m, candidates $\mathcal{C}$, threshold τ, min-keep $m_{\min}$, EM iters T
$s_i \leftarrow \text{score}(m, c_i)$ for each $c_i \in \mathcal{C}$
$x_i \leftarrow \text{minmax_to_unit}(s_i)$
if $\max s_i = \min s_i$ **then**
  $x_i \leftarrow 0.5$ for all i
end if
Initialize 2 components using quantiles $(q_{0.3}, q_{0.7})$ of $\{x_i\}$
for $t = 1$ to T **do**
  E-step: $r_{ik} \propto \pi_k \text{Beta}(x_i; \alpha_k, \beta_k)$ (computed in log-space)
  M-step: $\pi_k \leftarrow \frac{1}{n}\sum_i r_{ik}$; update (α_k, β_k) by weighted moments
end for
$k^\star \leftarrow \arg\max_k \frac{\alpha_k}{\alpha_k+\beta_k}$
$g_i \leftarrow p(z = k^\star \mid x_i)$ for all i
$\mathcal{I} \leftarrow \{i : g_i \geq \tau\}$
if $|\mathcal{I}| < m_{\min}$ **then**
  $\mathcal{I} \leftarrow \text{TopK}(\{x_i\}, m_{\min})$
end if
if $\mathcal{I} = \emptyset$ **then**
  return NEWSESSION(m)
end if
$s^\star \leftarrow \arg\max_{i \in \mathcal{I}} x_i$
return MERGE$(m, c_{s^\star})$

Thresholding with minimum-keep fallback. Given $\{g(x_i)\}$, we retain candidates with

$$g(x_i) \geq \tau, \tag{26}$$

where τ is a posterior threshold. If fewer than $m_{\min}$ candidates qualify, we retain the top-$m_{\min}$ by x_i, preserving the original score ranking.

Usage in memory fusion. The retained candidates define a filtered set of plausible merge targets. The incoming memory item is merged into the most compatible retained session (under the currently selected structure), or stored as a new session if no candidate is retained. After distribution fitting, the posterior threshold τ and minimum retention $m_{\min}$ explicitly control selection.

Computational complexity. For n candidate sessions, each EM iteration runs in $\mathcal{O}(n)$ time. Since the number of mixture components is fixed to two and the number of EM iterations is small and bounded, the overall gating cost is negligible compared to retrieval and response generation. The memory overhead is $\mathcal{O}(n)$ for storing normalized scores and responsibilities.

D Conversation Feature Definition for Structure Selection

D.1 Feature Overview

We extract a compact set of interpretable interaction-level features to characterize the structural properties of the ongoing conversation. These features are designed to capture temporal dynamics, entity-centricity, and topic evolution patterns, which are indicative of suitable memory organizations.

D.2 Feature Definition Table

As summarized in Table 5, these features are intentionally simple and interpretable, focusing on structural signals rather than deep semantic representations. This design keeps structure selection lightweight and stable, and avoids introducing additional reasoning overhead into the control layer.

Table 5: Lightweight structural features for conversation-level memory selection.

Feature	Description and Motivation
`page_count`	Number of interaction pages within the current context window. Captures interaction scale; longer conversations often benefit from structured or hierarchical memory.
`avg_page_length`	Average length of recent pages (in tokens or turns). Longer pages indicate information-dense interactions that may require non-linear organization.
`entity_density`	Average number of named entities per page. High values suggest entity-centric interactions, favoring graph-based memory.
`relation_indicators`	Frequency of explicit relational expressions (e.g., comparisons, references, dependencies). Signals the presence of relational structure beyond pure temporal order.
`topic_diversity`	Number of distinct topics detected in recent interactions. High diversity indicates branching or hierarchical discourse structure.
`topic_transitions`	Frequency of topic shifts across consecutive pages. Rapid transitions favor hierarchical or graph-based organization over linear memory.
`is_qna_pattern`	Binary indicator of question–answer dominated interaction patterns. Such patterns often align with linear, recency-driven memory access.
`is_decision_tree`	Binary indicator of decision-tree-like dialogue flow (e.g., successive conditional questions). Suggests hierarchical organization with coarse-to-fine retrieval.
`is_entity_centric`	Binary indicator of entity-focused queries (e.g., repeated mentions of the same entities). Favors graph-based memory structures.
`time_span`	Temporal span covered by recent interactions. Longer spans reduce the effectiveness of pure recency-based retrieval.
`temporal_density`	Interaction frequency per unit time. Dense interactions emphasize short-term dependencies and linear access.
`semantic_complexity`	Estimated semantic variability across recent pages (e.g., embedding dispersion). High complexity suggests heterogeneous content requiring structured organization.

E Structure Selector Training Pipeline

Offline supervision for structure selection. Directly optimizing memory structure selection via online reinforcement learning is undesirable in our setting, as it introduces instability, high variance, and additional interaction cost. Instead, we adopt an offline supervision strategy that derives structure labels from interaction-level outcomes. Specifically, we evaluate each candidate memory structure under identical interaction contexts and assign the structure that yields the highest empirical reward as the target label (Algorithm 2). This supervised formulation supports stable training grounded in downstream response quality (Algorithm 3).

Algorithm 2 Offline Structure Labeling via Interaction-Derived Rewards

Require: Training conversations $\mathcal{D}$, structures $\mathcal{S} = \{\text{LINEAR}, \text{GRAPH}, \text{HIERARCHICAL}\}$
Require: Fixed agent pipeline $\mathcal{A}(\cdot;\, s)$ instantiated with structure s
Require: Reward weights $\lambda_{\text{judge}}, \lambda_{\text{mem}}$ (e.g., 0.7/0.3)
Ensure: Labeled dataset $\mathcal{T} = \{(x, y)\}$

$\mathcal{T} \leftarrow \emptyset$
for each conversation/session instance $\xi \in \mathcal{D}$ **do**
  Extract feature vector $x \leftarrow \text{Feat}(\xi)$
  for each $s \in \mathcal{S}$ **do**
    Run agent with structure s: $\hat{a}_s \leftarrow \mathcal{A}(\xi;\, s)$
    Compute judge reward: $r_{\text{judge}}(s) \leftarrow \text{Judge}(\hat{a}_s, \xi)$
    Compute memory utilization reward: $r_{\text{mem}}(s) \leftarrow \text{MemUtil}(\hat{a}_s, \xi)$
    $r(s) \leftarrow \lambda_{\text{judge}}\, r_{\text{judge}}(s) + \lambda_{\text{mem}}\, r_{\text{mem}}(s)$
  end for
  $y \leftarrow \arg\max_{s \in \mathcal{S}} r(s)$
  (reward-optimal structure label)
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{(x, y)\}$
end for
return $\mathcal{T}$

Algorithm 3 Supervised Training of Structure Selector

Require: Labeled dataset $\mathcal{T} = \{(x_i, y_i)\}$, selector $f_\theta(\cdot)$ (MLP)
Require: Loss $\mathcal{L}$ (cross-entropy), optimizer ADAM
Ensure: Trained parameters θ

Fit feature normalizer (e.g., STANDARDSCALER) on $\{x_i\}$
for epoch $= 1$ to MaxEpoch **do**
  for each mini-batch $\mathcal{B} \subset \mathcal{T}$ **do**
    $\tilde{x} \leftarrow \text{Normalize}(x)$ for all $(x, y) \in \mathcal{B}$
    $p \leftarrow \text{Softmax}(f_\theta(\tilde{x}))$
    $\theta \leftarrow \theta - \eta \nabla_\theta \sum_{(x,y)\in\mathcal{B}} \mathcal{L}(p, y)$
  end for
  Early stop if validation performance does not improve
end for
return θ

F ADDITIONAL EXPERIMENTAL RESULTS

Table 6 reports category-wise results on LOCOMO using ROUGE-1, ROUGE-2, and BERTScore. Overall, **FluxMem** achieves the strongest and most balanced performance across all four reasoning categories. The improvements are particularly pronounced in multi-hop and single-hop reasoning, where FluxMem consistently attains the highest ROUGE and BERTScore values, indicating more accurate content selection and semantic alignment. On open-domain questions, FluxMem substantially outperforms baselines in ROUGE-1, reflecting its ability to retrieve relevant information under weak structural cues. While some baselines achieve competitive BERTScore in specific categories, they fail to maintain consistent performance across metrics and reasoning types. These results further confirm that adaptive multi-structure memory organization enables more robust retrieval and generation across diverse long-horizon reasoning scenarios.

G MEMORY STRUCTURE IMPLEMENTATIONS

For completeness, we briefly describe the implementation details of the three memory structures considered in Sec. 3.3. These structures are not proposed in this work but are adapted from prior studies on long-term memory for LLM agents.

Table 6: Performance comparison on LoCoMo across different reasoning categories. Best results are shown in bold, second-best are underlined.

Method	Multi-hop			Temporal		
	R-1	R-2	BS	R-1	R-2	BS
LangMem (LangChain, 2025)	<u>48.07</u>	13.80	65.16	**64.12**	21.55	80.87
Mem0 (Chhikara et al., 2025)	29.95	9.43	52.20	24.31	6.94	50.27
ZEP (Rasmussen et al., 2025)	8.00	2.11	32.38	6.56	1.06	19.05
MemoryOS (Kang et al., 2025)	44.27	<u>17.19</u>	<u>89.91</u>	44.87	19.84	<u>91.17</u>
A-Mem (Xu et al., 2025b)	34.54	13.62	54.95	40.86	19.01	65.67
MEMOS (Li et al., 2025b)	18.33	6.50	50.75	12.40	3.25	44.05
MemR^3 (Du et al., 2025)	17.34	6.20	–	24.15	8.00	–
O-mem (Wang et al., 2025)	36.83	14.53	30.81	50.02	**26.45**	50.38
HippoRAG 2 (Gutiérrez et al., 2025)	29.19	9.46	87.26	10.32	2.12	84.32
FluxMem	**49.80**	**19.34**	**90.39**	<u>56.96</u>	<u>23.95</u>	**92.81**

Method	Open			Single-hop		
	R-1	R-2	BS	R-1	R-2	BS
LangMem (LangChain, 2025)	24.55	<u>9.38</u>	45.19	<u>51.30</u>	29.15	68.49
Mem0 (Chhikara et al., 2025)	23.53	5.56	40.82	33.48	19.25	46.42
ZEP (Rasmussen et al., 2025)	6.10	0.92	34.09	6.94	2.26	26.29
MemoryOS (Kang et al., 2025)	<u>28.20</u>	**9.68**	<u>88.67</u>	50.77	<u>30.29</u>	<u>91.07</u>
A-Mem (Xu et al., 2025b)	16.85	3.66	40.06	43.06	25.66	56.13
MEMOS (Li et al., 2025b)	11.04	2.45	39.91	18.10	10.01	53.30
MemR^3 (Du et al., 2025)	10.39	2.26	–	23.09	13.77	–
O-mem (Wang et al., 2025)	21.62	3.19	26.84	45.16	27.74	41.60
HippoRAG 2 (Gutiérrez et al., 2025)	25.08	7.70	88.05	45.70	29.23	90.24
FluxMem	**37.42**	9.29	**89.47**	**63.83**	**38.98**	**92.58**

Linear Memory. Linear memory organizes episodic sessions in strict chronological order. Each memory unit is appended sequentially, and retrieval is primarily driven by recency or similarity-based ranking. This structure favors temporal coherence and recency.

Graph Memory. Graph memory represents episodic sessions as nodes in a graph, where edges encode semantic or entity-level relations (e.g., shared entities or high embedding similarity). Traversal of structurally related nodes supports multi-hop recall beyond temporal proximity.

Hierarchical Memory. Hierarchical memory clusters episodic sessions into higher-level abstractions (e.g., topics or summaries), forming a tree-like structure. Coarse-to-fine retrieval selects relevant clusters, then specific episodes, improving scalability and abstraction.

H Computational Cost Analysis

FluxMem adds memory selection and fusion costs while preserving backbone inference.

Memory structure selection. The memory structure selector is a two-layer MLP with a 12-dimensional input and hidden size 4, incurring negligible overhead compared to LLM inference. This module is executed once per interaction and adds a constant-time cost.

BMM-based memory fusion. The Beta Mixture Model (BMM) gate operates on matching scores of candidate memories and involves lightweight posterior computation. Fusion cost scales linearly with the retrieved MTEM candidates and is small in practice.

Retrieval cost. Retrieval combines dense encoding and BM25 via reciprocal rank fusion. Dense retrieval relies on pre-computed embeddings, and BM25 operates over indexed text, yielding costs comparable to standard hybrid LLM retrieval.

Overall cost. Compared to baselines with fixed memory structures, FluxMem introduces modest additional computation for structure selection and probabilistic gating. Empirically, this overhead

Table 7: Latency breakdown by component.

Component	Latency (ms)	FluxMem-specific?
LLM calls	∼2,860	No
Mid-term semantic search	∼800	Yes
Rerank	∼300	No
Embedding	∼32	No
MLP structure selector	0.05	Yes
BMM gating	0.6	Yes

Table 8: Inference-time cost comparison per query on LoCoMo.

Method	Total (ms)	PromptTok	CompTok	LLM	Disk
A-Mem	2583	11250	35	2	1.8MB
O-Mem	1956	1568	5	1	0.9MB
Mem0	1499	1461	5	1	cloud
MemoryOS	1980	1978	53	1	server
MemR^3	1807	3591	492	2.8	0.1MB
HippoRAG 2	2349	522	23	1	7.5MB
FluxMem	3990	4876	14	1	8.2MB

is minor relative to LLM decoding cost and does not significantly affect end-to-end latency, while enabling more adaptive and selective memory utilization.

Empirical latency comparison. Table 7 reports the per-component latency breakdown, and Table 8 compares inference-time cost per query across methods on LoCoMo. As shown in Table 7, FluxMem-specific components (MLP selector and BMM gating) add less than 1ms combined, accounting for less than 0.03% of total latency. The dominant cost remains LLM inference calls (69.6%), which is shared by all methods. While FluxMem's total latency in Table 8 is higher under this specific configuration (top-K=15 on GPT-4.1), Table 9 shows that FluxMem requires only 0.6 LLM calls per memory construction—the lowest among all methods—reducing offline cost.

Retrieval depth trade-off. Table 10 shows that reducing retrieval depth offers a practical knob for latency-sensitive deployments with minimal performance loss. Specifically, reducing top-K from 15 to 5 cuts latency by 39% (3980ms $\rightarrow$ 2419ms) with only 1.14 F1 loss, demonstrating that FluxMem can be flexibly configured for different latency budgets.

Cross-setting latency comparison. To provide a more complete picture, we report latency under different backbones and datasets. As shown in Tables 11 and 12, FluxMem is consistently among the fastest methods under diverse settings, demonstrating that the higher latency observed under GPT-4.1 with top-K=15 is configuration-specific rather than inherent.

I Homogeneous Ensemble Comparison

To verify that FluxMem's gains stem from structural diversity rather than ensemble redundancy, we compare against homogeneous configurations using three instances of the same structure type, each

Table 9: Memory construction cost per conversation on LoCoMo.

Method	LLM calls	Tokens	Mean (ms)
A-Mem	2.0	600	12033
O-Mem	1.5	750	6069
Mem0	1.0	400	–
HippoRAG 2	1.0	246	1698
FluxMem	0.60	357	5731

Table 10: Retrieval depth trade-off on LoCoMo (GPT-4.1).

Top-K	F1	Mean Latency (ms)
5	50.02	2419
10	50.41	3129
15	51.16	3980

Table 11: Latency on LoCoMo with Claude Haiku 4.5.

Method	Mean (ms)	Median (ms)
FluxMem	**2213**	**1962**
HippoRAG 2	3443	2932
MemoryOS	5587	3716
O-Mem	5774	3122
A-Mem	5401	4764

diversified through different retrieval configurations (e.g., varying expansion depth for Graph, ranking strategies for Linear, granularity levels for Hierarchy).

FluxMem outperforms the best homogeneous ensemble by a clear margin across all metrics, confirming that gains come from structural diversity rather than simple ensembling. No homogeneous configuration dominates, supporting structural complementarity in conversational memory.

J Cross-Backbone Evaluation

To verify generalization beyond a single LLM backbone, we additionally evaluate FluxMem and baselines using Claude Haiku 4.5. As shown in Table 14, FluxMem consistently achieves the best performance across all categories, demonstrating that the framework generalizes well across different LLM backbones without retraining the structure selector.

K Robustness and Variance Analysis

To assess the stability of reported results, we conduct repeated evaluations (3 runs) on LoCoMo. As shown in Table 15, all methods exhibit comparable variance, and FluxMem's improvements exceed one standard deviation of all baselines, indicating that the reported gains are stable across runs.

FluxMem's standard deviation matches or improves on baselines, indicating stability across runs.

L Reward Weighting Sensitivity

We analyze how the weighting between judge reward (λ_{judge}) and memory utilization reward (λ_{mem}) affects selector training and downstream performance. As shown in Table 16, using only judge reward ($\lambda_{\text{judge}} = 1.0$) significantly underperforms, indicating that memory utilization feedback provides complementary signals for structure selection. Conversely, equal weighting $(0.5, 0.5)$ also degrades performance, suggesting response quality should remain the dominant training signal.

Table 12: Latency on PERSONAMEM (GPT-4.1).

Method	Mean (ms)	Median (ms)
FluxMem	**1808**	**1465**
HippoRAG 2	2552	2177
MemoryOS	2797	2458
O-Mem	3872	3432
A-Mem	5102	5094

Table 13: Homogeneous ensemble vs. FluxMem on LoCoMo.

Method	F1	BLEU-1	ROUGE-L
3×Linear	46.12	36.04	46.37
3×Graph	42.42	38.00	42.59
3×Hierarchy	43.08	37.95	41.89
FluxMem	**51.20**	**41.23**	**49.51**

Table 14: LoCoMo results with Claude Haiku 4.5.

Method	Cat1: Multi-hop			Cat2: Temporal			Cat3: Open			Cat4: Single-hop		
	F1	B-1	R-L	F1	B-1	R-L	F1	B-1	R-L	F1	B-1	R-L
A-Mem	10.77	13.07	15.85	25.11	15.49	40.79	3.43	8.67	10.31	20.60	16.30	34.01
HippoRAG 2	14.60	11.26	12.21	12.64	8.42	12.39	12.91	12.01	12.07	40.76	28.42	38.00
MemoryOS	28.75	19.00	25.31	49.74	40.92	49.74	18.98	13.97	18.95	42.53	34.43	41.15
O-Mem	33.72	24.89	28.35	53.46	42.36	56.46	19.71	15.20	18.87	55.93	42.79	54.13
FluxMem	**34.72**	**24.31**	**28.97**	**56.26**	**45.32**	**56.26**	**20.71**	**14.86**	**20.05**	**61.23**	**48.88**	**58.97**

Table 15: Results with variance (mean ± std over 3 runs) on LoCoMo.

Method	F1	BLEU-1	ROUGE-L
A-Mem	31.69±0.79	26.87±0.64	32.10±0.39
MemR^3	20.49±0.46	11.89±0.56	17.49±0.77
HippoRAG 2	27.55±0.79	21.05±0.57	26.59±0.74
Mem0	27.35±0.70	21.41±0.62	26.21±0.65
FluxMem	**51.16±0.86**	**41.23±0.90**	**49.51±0.69**

Table 16: Sensitivity of reward weighting on LoCoMo.

$(\lambda_{\text{judge}}, \lambda_{\text{mem}})$	F1	BLEU-1	ROUGE-L
(1.0, 0.0)	43.80	33.24	42.98
(0.8, 0.2)	46.00	36.05	44.38
(0.7, 0.3)	**51.16**	**41.23**	**49.51**
(0.6, 0.4)	50.05	40.97	49.56
(0.5, 0.5)	47.91	38.58	47.50

The default weighting (0.7, 0.3) achieves the best results. Performance degrades when either component dominates, confirming both reward signals are necessary.

M Structure Selector Analysis

Fixed-pool routing pilot. We compare selection rules on frozen linear, graph, and hierarchical views of identical LoCoMo conversation content (Table 17). Nine conversations are split into five training, two validation, and two test conversations; each contributes prefixes of one and two sessions. The test set contains four selection decisions and eight questions. All policies share the same store, a four-page retrieval cap, a 1,024-token context cap, and one GPT-4.1 answer call per question. A new 12–4–3 MLP uses training-only feature normalization and validation-based checkpoint selection; seed 42 is prespecified as the primary run. Uniform random reports the exact expectation over the three cached outcomes, and the oracle selects by held-out composite reward (0.7 answer quality + 0.3 memory quality), not F1. The primary MLP gains 1.46 F1 points over random, while fixed hierarchy matches its performance and fixed linear attains the reward oracle. Thus, this short-prefix pilot does not establish an additional benefit from learned adaptation over fixed selection; it isolates routing with frozen pools rather than evaluating online memory updates.

Table 17: Fixed-pool routing pilot on held-out LoCoMo conversations. Accuracy accepts ties for the highest reward; the reward oracle is diagnostic.

Selection policy	F1 (%)	Reward	Selection accuracy (%)
MLP (seed 42, primary)	13.68	0.6529	75.00
Uniform random (expectation)	12.22	0.6303	66.67
Fixed linear	12.36	0.6531	100.00
Fixed graph	10.62	0.5849	25.00
Fixed hierarchy	13.68	0.6529	75.00
Feature heuristic	12.36	0.6531	100.00
Reward oracle	12.36	0.6531	100.00
MLP (seed 43)	12.36	0.6531	100.00
MLP (seed 44)	13.68	0.6496	50.00

Table 18: Retrieved structure distribution (%) on LoCoMo.

Category	Linear	Graph	Hierarchy
Multi-hop	16.8	31.3	51.9
Temporal	28.2	13.7	58.1
Open	7.6	44.3	48.1
Single-hop	11.8	14.6	73.6
Overall	15.4	20.6	63.9

Structure distribution across categories. Table 18 shows the proportion of retrieved episodic units assigned to each structure type per query category.

All three structures are meaningfully utilized. The distribution varies across categories, confirming that the selector actively adapts rather than defaulting to a single type.

Cross-structure retrieval. Table 19 shows that most queries retrieve multiple structure types.

Cross-backbone selector consistency. To verify the selector generalizes across LLM backbones, we compare predictions using GPT-4.1 vs. Llama-3.1 features on LoCoMo (1,887 sessions). The prediction agreement is 99.6%, suggesting structural rather than backbone-specific patterns.

N Case Studies

We present three representative cases (Figs. 4–6) to illustrate how FluxMem dynamically leverages linear, graph, and hierarchical memory structures for diverse long-horizon queries.

Time-dependent reasoning. As shown in Fig. 4, the first case asks when Melanie read a book, requiring alignment of a relative temporal expression with a session timestamp. Linear memory preserves chronology, allowing FluxMem to identify 2022; baselines miss the temporal link.

Complex relational reasoning. Fig. 5 presents a query about Caroline's place of origin, which requires integrating relocation history with a later country mention. FluxMem activates graph memory

Table 19: Proportion of queries retrieving from $\geq$2 structure types.

Category	Queries with $\geq$2 Types
Single-hop	63.4%
Multi-hop	67.0%
Temporal	63.2%
Open-domain	71.9%
Overall	64.8%

When did Melanie read the book "nothing is impossible"?

Coversation history:

.....

"session_date_time": "4:33 pm on 12 July, 2023",

"**speaker**": "Melanie",
"text": "Wow, Caroline! You're so inspiring for wanting to help others with their mental health. What's pushing you to keep going forward with it?"

"**speaker**": "Caroline",
"text": "I struggled with mental health, and support I got was really helpful. It made me realize how important it is for others to have a support system. So, I started looking into counseling and mental health career options, so I could help other people on their own journeys like I was helped."

"**speaker**": "Melanie",
"text": "Caroline, so glad you got the support! Your experience really brought you to where you need to be. You're gonna make a huge difference! This book I read **last year** reminds me to always pursue my dreams, just like you are doing!🌟"
book pic information :"a photography of a book cover with a gold coin on it"

"**speaker**": "Caroline",
"text": "Thanks so much, Mel! Seeing this pic just made me appreciate my love of reading even more. Books guide me, motivate me and help me discover who I am. They're a huge part of my journey, and this one's reminding me to keep going and **never give up**!"

Gold answer: 2022

Fluxmem: 2022 ✅

MemoryOS: No record ❌

Mem0: No evidence in memories ❌

Figure 4: Time-dependent query case.

to connect relational cues across sessions and correctly identifies Sweden, whereas competing methods fail to establish the necessary relations.

Topic evolution. In Fig. 6, the query appears in a conversation with substantial topic drift. By organizing episodic content hierarchically, FluxMem abstracts over evolving subtopics and retrieves the correct activity (horseback riding), avoiding interference from recent but irrelevant information.

O LLM PROMPTS DESIGN

Where did Caroline move from 4 years ago?

Coversation history:

.....

"session_date_time": "7:55 pm on 9 June, 2023",

"speaker": "Caroline",

"text": "Thanks, Mel! My friends, family and mentors are my rocks - they motivate me and give me the strength to push on. Here's a pic from when we met up last week!"

pic information: "a photo of a family posing for a picture in a yard"

"speaker": "Melanie",
"text": "Wow, that photo is great! How long have you had such a great support system?"

"speaker": "Caroline",

"text": "Yeah, I'm really lucky to have them. They've been there through everything, I've known these friends for 4 years, since **I moved from my home country**. Their love and help have been so important especially after that tough breakup. I'm super thankful. Who supports you, Mel?"

.....

"speaker": "Caroline",

"text": "Thanks, Melanie! This necklace is super special to me - a gift from my grandma in **my home country**, **Sweden**. She gave it to me when I was young, and it stands for love, faith and strength. It's like a reminder of my roots and all the love and support I get from my family.",

Gold answer: Sweden

Fluxmem: Sweden ✅

MemoryOS: Unknown ❌

Mem0: No information available ❌

Figure 5: Complex-relation case.

What activity did Caroline used to do with her dad?

Coversation history:

..... (a long conversation history about pets)

"speaker": "Melanie",

"text": "Oliver's hilarious! He hid his bone in my slipper once! Cute, right? Almost as silly as when I got to feed a horse a carrot. ",

"speaker": "Caroline",

"text": "That's so funny! I used to go horseback riding with my dad when I was a kid, we'd go through the fields, feeling the wind. It was so special. I've always had a love for horses!",

"speaker": "Melanie",

"text": "Wow, that sounds great - I agree, they're awesome. Here's a photo of my horse painting I did recently.",

Gold answer: Horseback riding

Fluxmem: horseback riding ✅

MemoryOS: drawing flowers ❌

Mem0: No information available ❌

Figure 6: Topic evolution case.

Prompt for Response

```
"<CONTEXT>\n"
    "Recent conversation between {user_a} and {user_b}:\n"
    "{history_text}\n\n"
    "<MEMORY>\n"
    "Relevant past conversations:\n"
    "{retrieval_text}\n"
    "<User Profile>\n"
    "Characteristics of {user_a}:\n"
    "{background}\n\n"
    "the question is: {query}\n"
    "Your task is to answer questions about {user_a} or {user_b} in an
extremely concise manner.\n"
```

Figure 7: Prompt for response.

Prompt for Meta_Info

```
"""Update the conversation meta-summary by incorporating the new dialogue
while maintaining continuity.
    Guidelines:
    1. Start from the previous meta-summary (if exists)
    2. Add/update information based on the new dialogue
    3. Keep it concise (1-2 sentences max)
    4. Maintain context coherence
    Previous Meta-summary: {last_meta}
    New Dialogue:
    {new_dialogue}
    Updated Meta-summary:"""
```

Figure 8: Prompt for Meta_Info.

Prompt for Entity Relation Extraction

```
"""Extract entities and relationships from the following conversation.
Conversation:
{conversation}
Output format (JSON):
{{"entities": [{{"id": "e1", "name": "Entity Name", "type":
"PERSON|ORGANIZATION|LOCATION|EVENT|TIME|ACTIVITY|OBJECT|CONCEPT|OTHER",
"properties": {{}}}}, ...],
  "relations": [{{"source": "e1", "target": "e2", "relation":
"relation_type", "weight": 1.0}}, ...]}}
Important:
- Extract ALL relevant entities including events, times, activities, objects,
and concepts
- For time entities, extract specific dates, months, years, or relative time
expressions
- For event entities, extract all mentioned events, meetings, activities
- For activity entities, extract hobbies, interests, and regular activities
- Extract relationships between entities when explicitly mentioned or clearly
implied
- Use concise names for entities
- If no entities found, return empty arrays:"""
```

Figure 9: Prompt for entity relation extraction.

Prompt for Hierarchical Structure Construction

```
"""Build a hierarchical topic tree from the following conversation.
Conversation:
{conversation}
Output format (JSON):
{{"root": {{
    "id": "root",
    "topic": "Main Topic",
    "children": [{{{
        "id": "t1",
        "topic": "Subtopic 1",
        "children": [],
        "keywords": ["keyword1", "keyword2"]
      }}, ...],
    "keywords": ["main", "keywords"]}}}}
Create a meaningful hierarchy. If topics are flat, use a single root with all
topics as direct children:"""
```

Figure 10: Prompt for hierarchical structure construction.

Prompt for Procedural extraction

```
"""Extract procedural / policy-like memories from the latest user-AI
conversation.
Conversation:
{conversation}
Output format (JSON array). Each item can be either a string policy, or an
object:
[
  {{
    "policy": "concise rule",
    "tags": ["response_style" | "preference" | "other"],
    "confidence": 0.0,
    "slot": "optional_attribute_name",
    "value": "optional_attribute_value"
  }}
]
Only include policies that help future behavior. Output ONLY JSON."""
```

Figure 11: Prompt for procedural extraction.