S-MEM: SPECIALIZED MEMORY AGENTS FOR LONG-TERM MEMORY OF LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term memory is essential for LLM agents operating over extended interactions, where answering a query may require connecting evidence scattered across sessions. Existing memory systems typically store past interactions in external structures, such as vector databases or knowledge graphs, which can fragment original interaction contexts and weaken contextual dependencies for long-horizon reasoning. Recent memory-agent approaches instead use dedicated agents whose contexts directly serve as memory, preserving interaction histories for local reasoning. However, they typically organize memory by individual interaction episodes, causing related experiences from different sessions to be stored in separate agents and making cross-session reasoning difficult. We propose **Specialized Memory Agents (S-MEM)**, a long-term memory framework that organizes memory agents by topic rather than by interaction episode. Each Specialist is dedicated to a specific topic and maintains the interaction history related to that topic across sessions. Given a query, the Master Agent activates relevant Specialists, which reason over their accumulated histories, and synthesizes their evidence for global reasoning. By consolidating related experiences within the same memory agent, S-MEM enables cross-session reasoning while preserving the contextual richness of original interactions. Experiments on LoCoMo and LongMemEval show that S-MEM consistently improves F1 and BLEU-1 over strong baselines.

1 INTRODUCTION

Large Language Model (LLM) agents are increasingly expected to operate over extended interactions spanning multiple sessions.

In such settings, answering a later query may require combining pieces of evidence that appeared in different earlier interactions (Maharana et al., 2024; Wu et al., 2025). Long-term memory is therefore essential for preserving relevant information across sessions and making it available for later reasoning. However, simply retaining long interaction histories is not sufficient: LLMs can struggle to use relevant information buried in long contexts, exhibiting lost-in-the-middle phenomenon (Liu et al., 2024). Therefore, supporting long-term interactions still requires dedicated memory systems for managing and using information from past sessions.

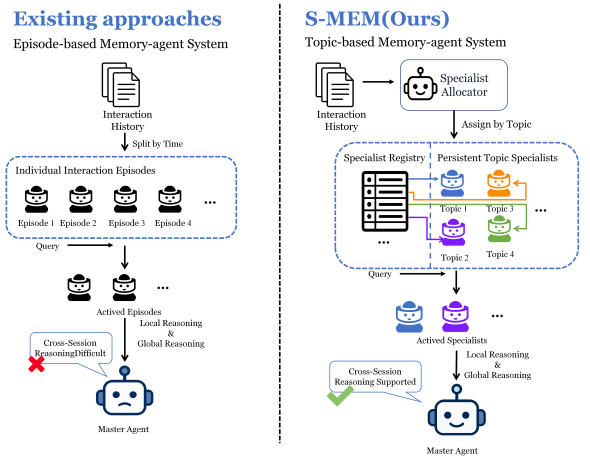

Figure 1: Comparison of episode-based memory-agent systems and S-MEM.

Most existing memory systems address this problem by storing past interactions in external memory structures, such as vector databases, knowledge graphs, or relational databases (Chhikara et al., 2025; Rasmussen

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.

et al., 2025; Hu et al., 2023), and retrieving relevant information when needed. While these structures make growing interaction histories easier and more efficient to store and retrieve, they can fragment information that was originally connected in the conversation. This fragmentation can weaken the contextual dependencies available during long-horizon reasoning (Wang et al., 2026; Zou et al., 2026; Tao et al., 2026).

A recent alternative is the *memory-agent* paradigm. In this paradigm, a *memory agent* is a dedicated agent whose interaction context directly serves as memory and can be reasoned over when that agent is activated. E-MEM is a representative work: it partitions an interaction stream into temporally bounded episodic contexts, assigns each context to a memory agent, and lets activated agents perform **local reasoning** within their own contexts before returning evidence to a central master agent (Wang et al., 2026). The master does not process the full interaction history directly; instead, it performs **global reasoning** by synthesizing the outputs of multiple memory agents. This design keeps each interaction episode intact within a memory's context, allowing the assistant to reason with the full local context rather than isolated memory fragments (Wang et al., 2026).

However, E-MEM is fundamentally an *episode-based memory-agent system*: each memory agent is associated with a temporally bounded episodic context (Wang et al., 2026). This preserves relationships within an episode, but long-term interactions are often organized around recurring topics rather than contiguous time. Information about the same topic may reappear after many subsequent sessions and therefore be distributed across different memory agents. Each memory agent can reason locally, but only over its own episode, which can make cross-session reasoning difficult. This raises a fundamental question: *how should memory agents be organized to support reasoning across sessions?*

As illustrated in Figure 1, our key idea is to organize memory agents around persistent topic specialization rather than temporally bounded interaction episodes. This idea is inspired by *Transactive Memory Systems* (TMS), where knowledge is distributed across specialized members, each responsible for a different area of expertise, and the group relies on knowing *who knows what* to access the right expertise when needed (Wegner, 1987; 1995). We adapt this principle to long-term memory for LLM agents: interactions concerning the same recurring topic are accumulated within the same memory agent across sessions, while a lightweight registry records which agent specializes in which topic. As a result, semantically related experiences that occur far apart in time can remain within a shared local reasoning context, rather than being separated simply because they occurred in different interaction episodes. At query time, the system can first identify *which Specialist knows about the relevant topic*, and then let that Specialist reason directly over its accumulated history. Based on this principle, we propose **Specialized Memory Agents (S-MEM)**, a *topic-based memory-agent system*. Each **Specialist** is a topic-specific memory agent that performs **local reasoning** over its accumulated history, while the **Master Agent** activates relevant Specialists and performs **global reasoning** by synthesizing their returned evidence and local deductions. By organizing memory agents around persistent topics, S-MEM preserves contextual dependencies across sessions, enabling more coherent and effective long-term reasoning. We conduct comprehensive experiments on two long-term memory benchmarks, LoCoMo (Maharana et al., 2024) and LongMemEval (Wu et al., 2025). Experimental results demonstrate that S-MEM consistently outperforms strong baselines across different backbone models, highlighting its effectiveness in supporting long-term reasoning.

Our contributions are as follows:

- We introduce a topic-specialized memory organization mechanism for LLM agents, where each Specialist maintains a persistent reasoning context that accumulates semantically related interactions across sessions. This design preserves contextual dependencies across temporally separated experiences, facilitating coherent long-term reasoning.
- We propose **S-MEM**, a topic-based memory-agent architecture that organizes long-term interaction histories around persistent Specialists rather than individual episodes. Each Specialist is a topic-specific memory agent whose accumulated interaction context directly serves as memory, allowing related experiences from different sessions to remain together for local reasoning.
- We conduct extensive evaluations on LoCoMo and LongMemEval, where S-MEM consistently outperforms strong baselines in F1 and BLEU-1, validating the effectiveness of our topic-based memory-agent design for long-term reasoning.

2 Related Work

Retrieval-Augmented Generation Retrieval-Augmented Generation (RAG) augments language models with information retrieved from external corpora (Lewis et al., 2020). Standard RAG typically partitions documents into retrievable units, indexes them in an external store, and selects relevant passages to support generation. Subsequent work improves this pipeline through more adaptive retrieval. FLARE dynamically determines when additional retrieval is needed during generation (Jiang et al., 2023), while Self-RAG jointly performs retrieval and self-reflection over retrieved evidence (Asai et al., 2024). Adaptive-RAG further selects different retrieval strategies according to query complexity, ranging from direct generation to multi-step retrieval (Jeong et al., 2024). Beyond retrieval control, recent work also explores richer mechanisms for organizing and accessing external knowledge. HippoRAG combines graph-structured knowledge with associative retrieval (Gutiérrez et al., 2024), while MemoRAG uses a global memory model to generate retrieval clues for downstream generation (Qian et al., 2025). However, organizing information into retrievable units can fragment context that was originally connected, which may weaken the contextual dependencies available for later reasoning (Wang et al., 2026; Fang et al., 2026).

Long-Term Memory for LLM Agents For LLM agents operating over extended interactions, long-term memory systems are needed to retain and organize information across sessions. Existing methods adopt diverse memory organizations (Yu et al., 2026; Huang et al., 2026; Kang et al., 2025; Wang & Chen, 2025; Latimer et al., 2026; Zhu et al., 2026). Generative Agents maintain experience streams with higher-level reflections (Park et al., 2023), MemoryBank maintains historical memories and evolving user information (Zhong et al., 2024), while Mem0 and LightMem consolidate conversational information into structured long-term memory (Chhikara et al., 2025; Fang et al., 2026). Other approaches employ richer structures: MemGPT uses multiple memory tiers (Packer et al., 2023), A-MEM constructs linked memory notes (Xu et al., 2025), Zep uses a temporal knowledge graph (Rasmussen et al., 2025), and GAM stores interaction history in a page-store for iterative search and integration (Yan et al., 2025). Despite their different organizations, these systems generally construct a memory store from past interactions and later access selected memory content for reasoning (Zhang et al., 2026; Jiang et al., 2026; Yang et al., 2026; Zhou & Han, 2025). These designs improve the organization and reuse of long-term memory, but reasoning still depends on the information exposed by the constructed memory structures rather than directly on the retained interaction context.

A recent alternative is the memory-agent paradigm, in which dedicated agents retain interaction histories directly within their contexts and perform local reasoning, thereby reducing information fragmentation caused by conventional memory extraction. E-MEM represents this paradigm by organizing memory agents around temporally bounded interaction episodes (Wang et al., 2026). In contrast, the proposed S-MEM organizes memory agents around persistent topics, allowing each Specialist to accumulate related interactions across sessions and reason over their shared context.

3 Methodology

We propose **S-MEM**, a topic-based memory-agent framework for long-term reasoning over multi-session interaction histories. Inspired by TMS, S-MEM organizes long-term memory around persistent topic-specific Specialists, allowing related experiences from different sessions to accumulate within the same local reasoning context. As illustrated in Figure 2, the framework consists of a **Master Agent**, a set of **Specialists**, and a **Specialist Allocator**. Its operation involves three stages: topic-based memory formation, Specialist activation, and hierarchical reasoning.

3.1 S-MEM Architecture

S-MEM consists of three main components: a Master Agent, a set of topic-specific Specialists, and a Specialist Allocator. Formally, the framework is defined as

$$\mathcal{F}_{\text{S-MEM}} = \left\langle A_{\text{master}}, \{A^{(i)}_{\text{spec}}\}_{i=1}^{N}, A_{\text{alloc}}, \mathcal{R} \right\rangle, \tag{1}$$

where A_{master} denotes the Master Agent, $A^{(i)}_{\text{spec}}$ denotes the i-th Specialist, and A_{alloc} denotes the Specialist Allocator. The Specialist Registry $\mathcal{R} = \{(z_i, d_i, A^{(i)}_{\text{spec}})\}_{i=1}^{N}$ maintains a mapping be-

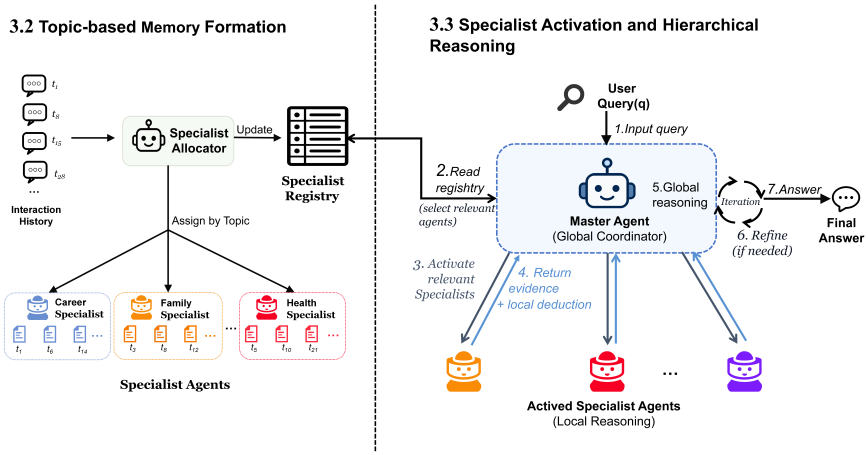

Figure 2: Overview of S-MEM. The Specialist Allocator organizes interaction histories into persistent topic-specific Specialists and maintains the Specialist Registry. Given a query, the Master Agent activates relevant Specialists, which perform local reasoning over their accumulated histories and return evidence and local deductions for global reasoning by the Master.

tween Specialists and their corresponding topics z_i and descriptions d_i. It supports Specialist activation and memory allocation without requiring direct access to the Specialists' complete interaction histories.

Master Agent: Specialist Activation and Global Reasoning. The Master Agent serves as the global coordinator of S-MEM. Rather than directly maintaining the interaction history, it identifies relevant Specialists through the Specialist Registry and activates them for local reasoning. It then synthesizes their outputs through global reasoning to answer a given query. Formally, the Master Agent is defined as

$$A_{\text{master}} = \langle f_{\text{activate}}, f_{\text{global}} \rangle , \tag{2}$$

where f_{activate} denotes the Specialist activation function, which identifies relevant Specialists based on the input query and the Specialist Registry, and f_{global} denotes the LLM-based global reasoning function, which integrates outputs of activate specialists to generate the answer. Together with the local reasoning performed by Specialists, these operations form the hierarchical reasoning process of S-MEM.

Specialist Agent: Topic-specific Memory and Local Reasoning. Each Specialist Agent is a topic-specific memory agent whose retained interaction history directly serves as its memory and reasoning context. Formally, the i-th Specialist is defined as

$$A^{(i)}_{\text{spec}} = \langle z_i, M_i, f_{\text{local}} \rangle , \tag{3}$$

where z_i denotes the Specialist's assigned topic, M_i represents its accumulated interaction history, and f_{local} denotes its LLM-based local reasoning function. Each Specialist retains interactions associated with its topic across different sessions, allowing temporally separated but semantically related experiences to accumulate within the same reasoning context. When activated, it reasons over its retained history and returns relevant evidence and local deductions to the Master Agent. Specialization arises from the topic-specific interaction history maintained by each Specialist rather than from a separately trained model.

Specialist Allocator: Memory Organization and Formation. The Specialist Allocator incrementally organizes incoming interactions into persistent topic-specific Specialists. It performs two complementary operations: assigning interactions to existing Specialists and creating new Specialists for previously uncovered topics. Formally, the Specialist Allocator is defined as

$$A_{\text{alloc}} = \langle f_{\text{assign}}, f_{\text{create}} \rangle , \tag{4}$$

where f_{assign} identifies relevant existing Specialists for incoming interactions, and f_{create} creates new Specialists when the existing ones do not adequately cover the topics in those interactions.

3.2 TOPIC-BASED MEMORY FORMATION

S-MEM incrementally transforms interaction histories into persistent topic-specific memories. Let the interaction history be $\mathcal{H}_T = (r_1, \ldots, r_T)$, where each record $r_t = (x_t, \tau_t, \mu_t)$ contains the original interaction text, timestamp, and metadata. We process $\mathcal{H}_T$ as overlapping interaction blocks $U_1, \ldots, U_J$:

$$U_j = (r_{s_j}, \ldots, r_{e_j}), \qquad j = 1, \ldots, J. \tag{5}$$

Each U_j consists of consecutive interaction records. Adjacent blocks retain a small overlap, i.e., $s_{j+1} \leq e_j$, to reduce semantic discontinuity introduced by block boundaries. Each U_j serves as the basic unit of memory allocation. The Specialist Allocator processes each incoming interaction block through two complementary operations: existing Specialist allocation and new Specialist formation.

Existing Specialist Allocation. Given an incoming interaction block U_j and the current Specialist Registry $\mathcal{R}$, the Allocator selects relevant existing Specialists based on the block's content and the Specialists' topic descriptions:

$$\mathcal{Z}_j = f_{\text{assign}}\left(U_j, \mathcal{R}\right), \tag{6}$$

where $\mathcal{Z}_j$ denotes the selected Specialist set. Since a block may involve multiple topics, it can be assigned to multiple Specialists. For each selected Specialist, U_j is appended to its retained interaction history:

$$M_i^{(j)} = M_i^{(j-1)} \| U_j, \qquad A_{\text{spec}}^{(i)} \in \mathcal{Z}_j. \tag{7}$$

Rather than extracting topic-specific facts from U_j, S-MEM stores the complete block in each selected Specialist, allowing related interactions from different sessions to accumulate within the same topic-specific history while preserving their contextual dependencies for subsequent reasoning.

New Specialist Formation. When an incoming interaction block contains topics that are not adequately covered by existing Specialists, the Allocator creates new Specialists and updates the Specialist Registry:

$$\mathcal{R}' = f_{\text{create}}\left(U_j, \mathcal{R}, \mathcal{Z}_j\right), \tag{8}$$

where $\mathcal{R}'$ denotes the updated Registry. f_{create} identifies topics that are not covered by the selected Specialists and, for each, creates a new Specialist initialized with the corresponding block and adds its topic and description to the Registry. If no topics remain uncovered, the Registry is unchanged.

The two operations are complementary: a block can be assigned to existing Specialists while also triggering new ones, allowing S-MEM to expand its memory organization while continuously accumulating related interactions within persistent Specialists.

3.3 SPECIALIST ACTIVATION AND HIERARCHICAL REASONING

At query time, S-MEM accesses long-term memory at the Specialist level rather than retrieving individual memory records. Given a query q, the Master Agent consults the Specialist Registry and activates relevant Specialists:

$$\mathcal{A}(q) = f_{\text{activate}}\left(q, \mathcal{R}\right), \qquad \mathcal{A}(q) \subseteq \{A_{\text{spec}}^{(i)}\}_{i=1}^{N}, \tag{9}$$

where $\mathcal{A}(q)$ denotes the activated Specialist set. The Registry provides compact descriptions of available Specialists, allowing the Master to locate relevant memory without accessing their complete interaction histories. Unlike conventional record-level retrieval, S-MEM activates memory-bearing agents capable of reasoning over their accumulated topic-specific contexts. A query may activate one or multiple Specialists depending on the topics involved.

Following Specialist activation, S-MEM performs **hierarchical reasoning** at two complementary levels. Activated Specialists perform **local reasoning** over their topic-specific histories, while the Master Agent performs **global reasoning** by integrating their outputs. For complex queries, this process can be repeated to refine generated results.

Local Reasoning. Each activated Specialist performs local reasoning over its retained interaction history to identify query-relevant evidence and derive local deductions. Given a query q, the local reasoning process is defined as

$$(E_i, D_i) = f_{\text{local}}(q, M_i), \qquad A^{(i)}_{\text{spec}} \in \mathcal{A}(q), \tag{10}$$

where E_i denotes the evidence identified from the Specialist's history and D_i denotes deductions derived from that evidence. Since M_i contains interactions accumulated under the same topic across sessions, the Specialist can connect temporally separated but related experiences within a shared reasoning context. Rather than merely returning stored information, each Specialist reasons directly over its retained history and returns relevant evidence and local deductions to the Master Agent.

Global Reasoning. After collecting the outputs of the activated Specialists, the Master Agent performs global reasoning to generate the answer:

$$\hat{y} = f_{\text{global}}\left(q, \{(E_i, D_i)\}_{A^{(i)}_{\text{spec}} \in \mathcal{A}(q)}\right), \tag{11}$$

where $\hat{y}$ denotes the generated answer. Unlike local reasoning, which is restricted to an individual Specialist's retained history, global reasoning integrates information across different topic-specific contexts. For queries involving a single topic, the output of one Specialist may be sufficient. For queries spanning multiple topics, the Master Agent combines evidence and local deductions from multiple Specialists to construct the answer.

Iterative Reasoning. For complex queries, the initially activated Specialists may not provide sufficient evidence. S-MEM therefore supports iterative reasoning, allowing the Master Agent to request additional information by activating other Specialists or revisiting previously activated ones. Let $\mathcal{A}_r$ denote the set of Specialists activated at reasoning round r, and let $\mathcal{O}_{\leq r}$ denote all Specialist outputs collected through reasoning round r. In iterative reasoning, the global reasoning function additionally evaluates evidence sufficiency and generates a refined information request when necessary:

$$(\hat{y}_r, b_r, q_{r+1}) = f_{\text{global}}(q, \mathcal{O}_{\leq r}), \qquad \mathcal{A}_{r+1} = f_{\text{activate}}(q_{r+1}, \mathcal{R}), \quad \text{if } b_r = 0. \tag{12}$$

where $\hat{y}_r$ denotes the current answer, b_r indicates whether the collected evidence is sufficient, and q_{r+1} denotes a refined information request when additional evidence is required. If the evidence is insufficient ($b_r = 0$), the Master Agent activates Specialists using the refined request and incorporates the newly returned evidence and local deductions into the next reasoning round. This process continues until the Master Agent determines that sufficient evidence has been collected or the reasoning process reaches its stopping condition.

Through hierarchical reasoning, S-MEM combines the contextual continuity of topic-specific Specialists with the Master Agent's ability to integrate information across different topics, supporting long-term reasoning over interactions distributed across multiple sessions.

4 Experiments

We evaluate S-MEM on long-term conversational question answering to assess its overall performance, validate its key architectural components, and investigate the effects of backbone selection across agent roles. Specifically, our experiments address three research questions: **RQ1:** How does S-MEM perform against existing long-term memory systems across different benchmarks and backbone models? **RQ2:** How do topic-based memory organization and Specialist local reasoning contribute to the effectiveness of S-MEM? **RQ3:** How sensitive is S-MEM to backbone selection across different agent roles?

4.1 Experimental Setup

Datasets. We evaluate S-MEM on two long-term memory benchmarks: **LoCoMo** (Maharana et al., 2024) and **LongMemEval-S** (Wu et al., 2025). LoCoMo is a benchmark for evaluating long-term conversational memory over multi-session dialogues. It tests five types of memory use: single-hop, multi-hop, temporal, open-domain, and adversarial questions. Its conversations average approximately 300 turns and 9K tokens and span up to 35 sessions. LongMemEval-S evaluates

Table 1: Results on LoCoMo benchmark. Bold and underline indicate the best and second-best results within each Master-backbone group, respectively.

Master	Method	Overall		Multi-Hop		Temporal		Open Domain		Single-Hop		Adversarial	
		F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
GPT-4o-mini	Full Context	35.25	29.11	14.70	10.34	19.93	12.72	8.06	5.28	25.09	15.12	84.30	84.30
	Naïve RAG	53.89	45.46	29.67	18.38	49.78	32.32	21.42	14.08	52.57	40.80	81.44	81.42
	LightMem	58.84	49.62	37.00	25.71	57.39	43.46	27.17	19.35	52.89	41.14	91.70	91.70
	GAM	60.97	55.96	43.23	34.90	57.60	45.88	**35.30**	**29.68**	57.57	51.73	86.55	86.55
	A-MEM	43.97	38.13	23.71	17.26	37.66	33.17	11.74	9.99	35.43	29.55	84.22	77.00
	E-MEM	64.04	54.84	40.35	31.82	55.17	40.28	32.30	25.89	60.64	52.96	96.85	89.66
	S-MEM	**67.57**	**60.00**	**47.61**	**39.67**	**61.95**	**47.90**	30.75	24.05	**64.92**	**55.91**	**97.16**	**97.01**
Qwen3.5-27B	Full Context	65.03	56.24	45.94	35.18	57.45	45.59	31.70	27.14	63.60	51.51	92.43	92.40
	Naïve RAG	58.33	50.51	31.54	22.44	54.68	43.72	19.30	14.83	54.24	43.55	94.00	93.97
	LightMem	62.67	57.13	41.34	31.94	63.78	57.53	23.85	19.46	55.91	48.89	96.45	96.43
	GAM	67.72	61.70	47.69	33.99	64.96	**58.36**	35.66	28.85	62.14	56.19	97.78	97.78
	A-MEM	50.80	46.34	33.90	24.24	37.35	32.26	11.79	9.31	43.00	37.97	94.29	94.14
	E-MEM	65.89	58.56	40.79	32.37	60.92	45.08	34.27	27.26	63.58	56.03	96.49	96.32
	S-MEM	**69.60**	**62.19**	**48.18**	**38.52**	**65.51**	53.00	**38.60**	**31.77**	**66.39**	**58.09**	**98.08**	**98.05**

long-term memory over substantially longer interaction histories, and covers information extraction, multi-session reasoning, knowledge updates, temporal reasoning, and abstention. Its histories average approximately 115K tokens across 40 sessions, making it a more challenging setting for locating and reasoning over information distributed across extended interactions.

Baselines. We compare S-MEM with **Full Context** and **Naïve RAG** (Lewis et al., 2020), together with recent long-term memory systems including **A-MEM** (Xu et al., 2025), **LightMem** (Fang et al., 2026), **GAM** (Yan et al., 2025), and **E-MEM** (Wang et al., 2026). These baselines span direct long-context inference, retrieval-augmented generation, structured external memory, and memory-agent systems.

Implementation. We evaluate GPT-4o-mini and Qwen3.5-27B as Master backbones. Unless otherwise specified, DeepSeek-V4-Flash serves as the backbone for the Specialist Allocator and Specialists in S-MEM, as well as for the memory agents in E-MEM. Main experiments use 4,000-token allocation blocks with overlapping context between adjacent blocks. We report F1 and BLEU-1; prompts and further implementation details appear in Appendix A.

4.2 Overall Performance Across Benchmarks and Backbones

LoCoMo. Table 1 shows that S-MEM achieves the best overall performance under both Master backbones, surpassing the strongest competing result in overall F1 by 3.53% with GPT-4o-mini and 1.88% with Qwen3.5-27B. The gains are particularly clear on multi-hop and temporal reasoning. Compared with E-MEM, S-MEM improves multi-hop F1 from 40.35% to 47.61% and temporal F1 from 55.17% to 61.95% with GPT-4o-mini, with similar gains under Qwen3.5-27B. These results reflect the difference in memory organization. E-MEM preserves context within temporally bounded episodes, whereas S-MEM accumulates interactions concerning the same recurring topic within persistent Specialists. This keeps cross-session evidence in a shared local reasoning context, making it easier to connect temporally separated information. This provides a plausible explanation for the observed improvements in multi-hop and temporal reasoning. Beyond these categories, S-MEM also achieves the highest single-hop and adversarial F1 among all evaluated baselines under both Master backbones. Together, these results suggest that topic-specialized memory organization supports not only complex cross-session reasoning but also a broader range of long-term memory tasks.

Table 2: Results on LongMemEval-S.

Method	GPT-4o-mini F1	GPT-4o-mini BLEU-1	Qwen3.5-27B F1	Qwen3.5-27B BLEU-1
Full Context	36.44	25.90	43.69	33.75
Naïve RAG	55.30	42.73	50.27	38.77
LightMem	32.81	23.21	36.76	26.13
GAM	39.62	32.54	53.37	45.68
A-MEM	56.65	44.98	60.83	51.12
E-MEM	42.09	32.14	58.59	48.35
S-MEM	**57.53**	**50.39**	**61.29**	**57.59**

LongMemEval-S. We further evaluate whether S-MEM's advantage extends to the substantially longer interaction histories in LongMemEval-S. As shown in Table 2, S-MEM again achieves the highest overall F1 and BLEU-1 under both Master backbones. Compared with A-MEM, the strongest com-

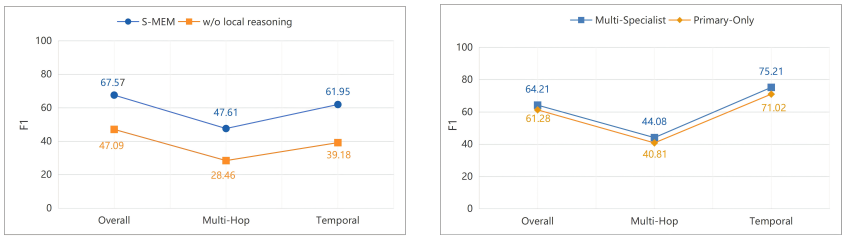

Figure 3: Analysis of core S-MEM design choices. Left: removing local reasoning from Specialists substantially degrades performance. Right: allowing multi-Specialist assignment improves performance over primary-only assignment.

peting system in overall F1, S-MEM improves F1 by 0.88% and 0.46%, and BLEU-1 by 5.41% and 6.47% with GPT-4o-mini and Qwen3.5-27B, respectively. The contrast with Full Context is particularly notable. With Qwen3.5-27B, Full Context reaches 65.03% F1 on LoCoMo but only 43.69% on LongMemEval-S, compared with 69.60% and 61.29% for S-MEM. This suggests that directly locating and integrating evidence becomes increasingly difficult as interaction histories grow longer. By separating topic-level local reasoning from global synthesis, S-MEM remains effective in the substantially more demanding long-history setting.

4.3 Effectiveness of Topic-Based Memory and Local Reasoning

We next examine two central aspects of S-MEM: whether topic-based memory formation preserves the evidence required for later reasoning, and whether Specialists benefit from reasoning locally over their accumulated histories. We additionally analyze multi-Specialist assignment.

Quality of Topic-based Memory Organization. We first evaluate whether the Specialist Allocator reliably organizes incoming interactions. **Evidence Reachability** measures whether all evidence required for a question remains accessible through the resulting Specialists. **Write Routing Precision** and **Write Routing Recall** measure the correctness and coverage of topic assignments, while **Avg. Routes/Record** reflects routing redundancy.

Table 3: Analysis of Specialist allocation on the first LoCoMo conversation. Reach. denotes Evidence Reachability, P/R denote Write Routing Precision/Recall, and Routes denotes Avg. Routes/Record.

Allocator	Reach.	P	R	Routes
GPT-4o-mini	97.46	97.73	75.30	1.82
Qwen3.5-27B	**99.49**	95.45	**85.15**	2.08
DeepSeek-V4-Flash	98.98	**100.00**	75.75	2.55

As shown in Table 3, all three Allocators achieve high Evidence Reachability, ranging from 97.46% to 99.49%, despite clear differences in routing behavior. Qwen3.5-27B achieves the highest reachability and routing recall, whereas DeepSeek-V4-Flash attains 100.00% routing precision while assigning more Specialists per record. The consistently high reachability indicates that the resulting topic-based organization retains nearly all evidence required by the evaluated questions, although different Allocator backbones produce different assignment patterns.

Contribution of Specialist Local Reasoning. We next examine whether topic-based organization alone is sufficient. We remove local reasoning from the Specialists and instead pass their retained histories directly to the Master. As shown in Figure 3 (left), overall F1 drops from 67.57% to 47.09%. Multi-hop and temporal F1 decrease from 47.61% to 28.46% and from 61.95% to 39.18%, respectively. The substantial degradation shows that organizing interactions by topic is not sufficient by itself. Specialists must reason over their accumulated histories to identify relevant evidence and local deductions before passing their outputs to the Master. This supports their role as active reasoning components rather than passive memory containers.

Effect of Multi-Specialist Assignment. We investigate whether assigning an interaction block to multiple Specialists is beneficial for preserving information across different topics. We compare two assignment strategies: primary-only assignment, which assigns each interaction block exclusively

to its most relevant Specialist, and multi-Specialist assignment, which allows the same block to be assigned to all relevant Specialists. As shown in Figure 3 (right), on the first two LoCoMo conversations, multi-Specialist assignment improves overall F1 from 61.28% to 64.21%. Multi-hop F1 increases from 40.81% to 44.08%, while temporal F1 increases from 71.02% to 75.21%. These gains suggest that exclusive assignment can leave information related to secondary topics unavailable in the Specialist activated later. Multi-Specialist assignment helps retain such information across the relevant topic-specific reasoning contexts. Complete category-level results for both ablations are provided in Appendix B.

4.4 Backbone Sensitivity Across Agent Roles

Finally, we examine how backbone selection affects agent roles in S-MEM. We use Qwen3.5-27B, DeepSeek-V4-Flash, and DeepSeek-V4-Flash as backbones for the Master Agent, Specialist Allocator, and Specialist, respectively, varying one role at a time while keeping the others fixed.

As shown in Table 4, backbone selection affects the three roles differently. Varying the Master changes overall F1 only from 64.83% to 66.61%, while varying the Specialist Allocator results in a similarly small range of 66.29%–67.79%. In contrast, changing the Specialist backbone produces a much larger range, from 54.99% to 68.61%.

Table 4: Role-wise backbone sensitivity on the first LoCoMo conversation. Only one role is varied at a time.

Varied Role	Backbone	F1	BLEU-1
Master	GPT-4o-mini	64.83	57.30
	Qwen3.5-27B	66.41	**60.65**
	DeepSeek-V4-Flash	**66.61**	59.34
Allocator	GPT-4o-mini	67.10	60.42
	Qwen3.5-27B	66.29	60.18
	DeepSeek-V4-Flash	**67.79**	**61.39**
Specialist	GPT-4o-mini	54.99	49.00
	Qwen3.5-27B	**68.61**	**61.03**
	DeepSeek-V4-Flash	66.41	60.65

The small variation across **Master Agent** backbones supports the hierarchical reasoning design of S-MEM. Since Specialists first identify relevant evidence and derive local deductions, the Master can focus on global synthesis rather than reasoning directly over long interaction histories. The **Specialist Allocator** shows a similar level of stability. This is consistent with Table 3: although different backbones produce different routing behavior, all preserve more than 97% Evidence Reachability. In contrast, **Specialist** performance is substantially more sensitive to backbone selection because it must directly reason over an accumulated topic history, identify query-relevant evidence, and derive local deductions. This is consistent with the local-reasoning ablation in Figure 3, where removing Specialist reasoning causes a substantial performance drop.

Together, these results support S-MEM's hierarchical division of labor: the Allocator organizes memory reliably, Specialists perform the main local reasoning over topic histories, and the Master focuses on global synthesis. The Specialist's greater sensitivity further indicates that delegating local memory reasoning to topic-specific Specialists is key to S-MEM's effectiveness. Complete category-level results appear in Appendix D, with additional analyses on allocation block size and computational cost in Appendices C and E, respectively.

5 Conclusion

In this work, we proposed S-MEM, a long-term memory framework that organizes interaction histories into persistent topic-specific Specialists rather than temporally bounded episodes. By accumulating related interactions across sessions, each Specialist maintains a coherent topic-specific reasoning context, while the Master Agent coordinates relevant Specialists and integrates their local evidence and deductions for global reasoning. Experiments on LoCoMo and LongMemEval-S demonstrate consistent improvements in overall F1 and BLEU-1 across different Master backbones, with notable gains in multi-hop and temporal reasoning. Further analyses show that S-MEM preserves high evidence reachability during memory formation and that Specialist local reasoning is essential to its effectiveness. Our backbone sensitivity analysis also reveals that performance is particularly sensitive to the Specialist backbone, highlighting the importance of local reasoning over accumulated topic histories. Together, these findings demonstrate the potential of topic-specialized memory agents for supporting long-term reasoning across multiple interaction sessions.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing, including improving grammatical correctness, clarity, and presentation. All AI-assisted revisions were reviewed and edited by the authors to ensure technical accuracy and consistency with the intended scientific claims. The authors take full responsibility for the final content of this work, including any text produced or revised with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide methodological and experimental details to support reproduction of our results. Section 3 specifies the S-MEM architecture, topic-based memory formation, Specialist activation, and hierarchical reasoning procedures, while Section 4.1 describes the datasets, baselines, model backbones, evaluation metrics, and main experimental configuration. Appendix A provides additional experimental configurations and evaluation details, and Appendix F provides the core prompt templates and inference settings used by S-MEM. We further report complete ablation results, allocation-block-size sensitivity, role-wise backbone results, and computational cost in Appendices B, C, D, and E, respectively. These materials document the main configurations and evaluation procedures used in our experiments.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, Vienna, Austria, May 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

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

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. LightMem: Lightweight and efficient memory-augmented generation. In *The Fourteenth International Conference on Learning Representations*, Rio de Janeiro, Brazil, April 2026. URL https://openreview.net/forum?id=dyJ0GWpjJB.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 59532–59569. Curran Associates, Inc., December 2024. doi: 10.52202/079017-1902. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/6ddc001d07ca4f319af96a3024f6dbd1-Paper-Conference.pdf.

Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. ChatDB: Augmenting LLMs with databases as their symbolic memory, 2023. URL https://arxiv.org/abs/2306.03901.

Weiquan Huang, Zixuan Wang, Hehai Lin, Sudong Wang, Bo Xu, Qian Li, Beier Zhu, Linyi Yang, and Chengwei Qin. AMA: Adaptive memory via multi-agent collaboration. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 3099–3120, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.152. URL https://aclanthology.org/2026.findings-acl.152/.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7036–7050, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.389. URL https://aclanthology.org/2024.naacl-long.389/.

Dongming Jiang, Yi Li, Guanpeng Li, and Bingzhe Li. MAGMA: A multi-graph based agentic memory architecture for AI agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36848–36865, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1709. URL https://aclanthology.org/2026.acl-long.1709/.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.495. URL https://aclanthology.org/2023.emnlp-main.495/.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25961–25970, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1318. URL https://aclanthology.org/2025.emnlp-main.1318/.

Christopher Latimer, Nicolò Boschi, Andrew Neeser, Chris Bartholomew, Gaurav Srivastava, Xuan Wang, and Naren Ramakrishnan. Hindsight: Structured agent memory that retains, recalls, and reflects. In Greg Durrett and Ping Jian (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 275–285, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-392-0. doi: 10.18653/v1/2026.acl-demo.27. URL https://aclanthology.org/2026.acl-demo.27/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., December 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In Lun-Wei Ku, André Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

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

Joon Sung Park, Joseph O'Brien, Carrie Jun 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 '23, pp. 1–22. ACM, October 2023. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. MemoRAG: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM Web Conference 2025*, WWW '25, pp. 2366–2377. ACM, April 2025. doi: 10.1145/3696410.3714805. URL https://doi.org/10.1145/3696410.3714805.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory, 2025. URL https://arxiv.org/abs/2501.13956.

Dehao Tao, Guoliang Ma, Yongfeng Huang, and Minghu Jiang. Membox: Weaving topic continuity into long-range memory for LLM agents, 2026. URL https://arxiv.org/abs/2601.03785.

Kaixiang Wang, Yidan Lin, Zihan Wang, Bunyod Suvonov, Zhaojiacheng Zhou, Yuxiang Zheng, Jiaxi Cao, Zhiheng Dong, Chentao Wu, Jiong Lou, and Jie Li. E-mem: Multi-agent based episodic context reconstruction for LLM agent memory. In *Proceedings of the 43rd International Conference on Machine Learning*, Seoul, South Korea, July 2026. PMLR. URL https://arxiv.org/abs/2601.21714.

Yu Wang and Xi Chen. MIRIX: Multi-agent memory system for LLM-based agents, 2025. URL https://arxiv.org/abs/2507.07957.

Daniel M. Wegner. Transactive memory: A contemporary analysis of the group mind. In Brian Mullen and George R. Goethals (eds.), *Theories of Group Behavior*, pp. 185–208. Springer, New York, NY, 1987. doi: 10.1007/978-1-4612-4634-3_9.

Daniel M. Wegner. A computer network model of human transactive memory. *Social Cognition*, 13 (3):319–339, 1995. doi: 10.1521/soco.1995.13.3.319.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations*, Singapore, April 2025. URL https://openreview.net/forum?id=pZiyCaVuti.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-Mem: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, volume 38, pp. 20004–20031. Curran Associates, Inc., December 2025. doi: 10.52202/085713-0593. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/19909c36f51abc4856b4560aff3d36d6-Paper-Conference.pdf.

B. Y. Yan, Chaofan Li, Hongjin Qian, Shuqi Lu, and Zheng Liu. General agentic memory via deep research, 2025. URL https://arxiv.org/abs/2511.18423.

Ruozhen Yang, Yucheng Jiang, Yueqi Jiang, Priyanka Kargupta, Yunyi Zhang, and Jiawei Han. Grounding agent memory in contextual intent. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 12008–12042, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.584. URL https://aclanthology.org/2026.findings-acl.584/.

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. Agentic memory: Learning unified long-term and short-term memory management for large language model agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21457–21483, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.981. URL https://aclanthology.org/2026.acl-long.981/.

Xiaohui Zhang, Zequn Sun, Chengyuan Yang, Yaqin Jin, Yazhong Zhang, and Wei Hu. ActMem: Bridging the gap between memory retrieval and reasoning in LLM agents, 2026. URL https://arxiv.org/abs/2603.00026.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19724–19731, March 2024. doi: 10.1609/aaai.v38i17.29946. URL https://doi.org/10.1609/aaai.v38i17.29946.

Sizhe Zhou and Jiawei Han. A simple yet strong baseline for long-term conversational memory of LLM agents, 2025. URL https://arxiv.org/abs/2511.17208.

Jinchang Zhu, Jindong Li, Cheng Zhang, Jiahong Liu, and Menglin Yang. HeLa-mem: Hebbian learning and associative memory for LLM agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13757–13769, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.625. URL https://aclanthology.org/2026.acl-long.625/.

Huhai Zou, Tianhao Sun, Chuanjiang He, Yu Tian, Zhenyang Li, Li Jin, Nayu Liu, Jiang Zhong, and Kaiwen Wei. ES-Mem: Event segmentation-based memory for long-term dialogue agents, 2026. URL https://arxiv.org/abs/2601.07582.

Table 5: Experimental configurations used in the main evaluation and additional analyses. M, A, and S denote the Master, Specialist Allocator, and Specialist, respectively. GPT denotes GPT-4o-mini, Qwen denotes Qwen3.5-27B, DS denotes DeepSeek-V4-Flash, and LME-S denotes LongMemEval-S. C1 and C1–2 denote the first one and two LoCoMo conversations, respectively.

Setting	Data	M	A	S
Main	LoCoMo / LME-S	GPT / Qwen	DS	DS
Local reasoning	LoCoMo	GPT	DS	DS
Memory assignment	C1–2	GPT	DS	DS
Block size	C1	GPT	DS	DS
Role backbone	C1	Varied	Varied	Varied
Efficiency	LoCoMo	GPT	DS	DS

Table 6: Complete local-reasoning ablation on the full LoCoMo benchmark. GPT-4o-mini is used as the Master and DeepSeek-V4-Flash is used for the Specialist Allocator and Specialists.

Variant	Overall		Multi-Hop		Temporal		Open Domain		Single-Hop		Adversarial	
	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
S-MEM	**67.57**	**60.00**	**47.61**	**39.67**	**61.95**	**47.90**	**30.75**	**24.05**	**64.92**	**55.91**	**97.16**	**97.01**
w/o Local Reasoning	47.09	41.72	28.46	20.72	39.18	29.08	20.78	16.45	44.39	38.59	75.32	75.45

A Additional Experimental Details

This section provides additional experimental details and the complete results omitted from the main text for space.

A.1 Experimental Configurations

Table 5 summarizes the configurations used in the main evaluation and additional analyses. The main experiments use DeepSeek-V4-Flash for both the Specialist Allocator and Specialists, while GPT-4o-mini and Qwen3.5-27B are evaluated as Master backbones.

The main S-MEM experiments use 4,000-token allocation blocks. This configuration was fixed before the block-size sensitivity experiment in Section C.

For the routing analysis in Table 3, semantically equivalent topic assignments are normalized into canonical topics using GPT-4.1-mini before computing Write Routing Precision and Recall. Both metrics are macro-averaged across records. Evidence Reachability is computed at the question level: a question is counted as reachable only when all evidence required for answering it remains accessible through the resulting Specialist organization.

B Complete Ablation Results

The main text reports compact results for the two core design ablations. Here we provide their complete category-level results.

B.1 Local Reasoning

Table 6 expands the local-reasoning ablation in Figure 3 to all LoCoMo question categories.

B.2 Multi-Specialist Assignment

Table 7 gives the complete results corresponding to the assignment analysis in Figure 3. Multi-Specialist assignment improves most categories on the evaluated subset, while the adversarial category is an exception, where primary-only assignment performs better.

Table 7: Complete comparison between multi-Specialist and primary-only assignment on the first two LoCoMo conversations.

Assignment	Overall		Multi-Hop		Temporal		Open Domain		Single-Hop		Adversarial	
	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
Multi-Specialist	**64.21**	**56.05**	**44.08**	**39.39**	**75.21**	**61.75**	**29.79**	**13.61**	**51.46**	**41.52**	93.43	92.18
Primary-only	61.28	53.91	40.81	37.42	71.02	58.14	28.01	13.07	45.29	36.14	**96.82**	**96.14**

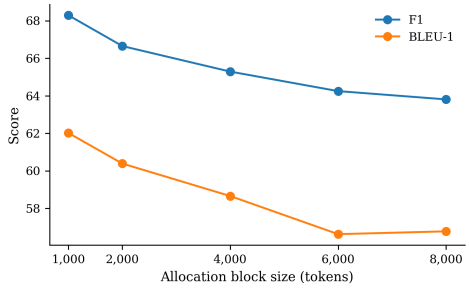

Figure 4: Sensitivity to allocation block size on the first LoCoMo conversation. The main experiments use 4,000-token blocks.

C Sensitivity to Allocation Block Size

We vary the allocation block size from 1,000 to 8,000 tokens on the first LoCoMo conversation. The 4,000-token setting used in the main experiments was fixed before this sensitivity analysis was conducted.

As shown in Figure 4, the overall performance is strongest with the smallest allocation blocks in this experiment. Table 8 provides the complete category-level results. The effect is not uniform across question types: for example, multi-hop F1 peaks at 2,000 tokens, whereas open-domain F1 is highest at 6,000 tokens.

D Complete Role-wise Backbone Results

Table 9 expands the role-wise backbone analysis in Table 4 to all LoCoMo question categories. Only one architectural role is varied at a time.

Although the aggregate Master and Allocator results are relatively stable, the category-level results show that different backbones can favor different forms of reasoning. The substantially larger variation among Specialist backbones, however, persists across multiple categories, consistent with the stronger model sensitivity observed in the main text.

E Efficiency and Computational Cost

Table 10 reports average latency and token consumption per query on LoCoMo under the GPT-4o-mini setting.

S-MEM uses substantially more tokens than the other evaluated methods, primarily because activated Specialists reason over their accumulated topic histories rather than a small set of retrieved records. In contrast, the increase in wall-clock latency is considerably smaller: S-MEM requires 19.96 seconds per query, close to GAM at 19.71 seconds and moderately above E-MEM at 17.02 seconds. The current implementation therefore trades additional memory-side computation for stronger long-term reasoning performance.

Table 8: Complete allocation-block-size sensitivity results on the first LoCoMo conversation. Token counts are decimal counts. The 4,000-token row corresponds to the final experimental configuration.

Tokens	Overall		Multi-Hop		Temporal		Open Domain		Single-Hop		Adversarial	
	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
1,000	**68.29**	**62.02**	46.78	40.57	**85.17**	**79.47**	32.76	**19.23**	55.93	**46.48**	**97.87**	**97.87**
2,000	66.65	60.39	**50.69**	**45.18**	76.85	71.46	28.69	13.98	**56.06**	46.21	95.74	95.98
4,000	65.29	58.66	46.69	41.12	72.22	66.41	29.06	15.22	54.99	44.31	**97.87**	**97.87**
6,000	64.25	56.63	45.59	37.00	75.17	68.20	**33.80**	15.26	51.53	40.90	95.74	95.74
8,000	63.81	56.78	45.31	40.57	74.70	66.26	30.16	15.95	52.76	41.87	93.62	93.85

Table 9: Complete role-wise backbone analysis on the first LoCoMo conversation.

Varied Role	Backbone	Overall		Multi-Hop		Temporal		Open Domain		Single-Hop		Adversarial	
		F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1	F1	BLEU-1
Master	GPT-4o-mini	64.83	57.30	46.61	**41.74**	69.07	60.65	29.79	13.61	58.10	46.38	93.62	93.62
	Qwen3.5-27B	66.41	**60.65**	43.67	37.91	**82.77**	**76.77**	25.48	**16.44**	57.35	**48.47**	**93.84**	93.80
	DeepSeek-V4-Flash	**66.61**	59.34	**49.23**	41.60	73.34	68.61	**37.13**	15.36	**58.35**	47.54	93.62	**93.85**
Allocator	GPT-4o-mini	67.10	60.42	40.29	33.04	**84.23**	**78.30**	**32.79**	18.06	57.44	47.64	**95.74**	**95.74**
	Qwen3.5-27B	66.29	60.18	44.26	36.80	80.45	74.32	23.05	17.17	57.12	47.50	**95.74**	**95.74**
	DeepSeek-V4-Flash	**67.79**	**61.39**	**45.57**	**38.80**	82.45	75.73	29.23	**20.68**	**59.84**	**49.94**	93.90	93.79
Specialist	GPT-4o-mini	54.99	49.00	37.43	31.77	65.41	56.31	20.04	7.83	58.89	**51.44**	62.62	62.72
	Qwen3.5-27B	**68.61**	**61.03**	**51.96**	**42.50**	78.90	71.12	23.55	13.78	**60.94**	49.64	**95.74**	**95.74**
	DeepSeek-V4-Flash	66.41	60.65	43.67	37.91	**82.77**	**76.77**	**25.48**	**16.44**	57.35	48.47	93.84	93.80

Table 10: Average latency and token consumption per query on LoCoMo with GPT-4o-mini. Role-specific token counts are shown where available.

Method	Time (s)	Avg. Tokens	Master Tokens	Memory Tokens
Full Context	2.32	33,386.71	–	–
Naïve RAG	3.87	4,186.67	–	–
LightMem	2.83	3,168.66	–	–
GAM	19.71	22,524.96	–	–
A-MEM	5.53	8,122.58	–	–
E-MEM	17.02	31,383.08	518.05	30,612.97
S-MEM	19.96	113,388.60	10,988.02	101,596.20

F Reproducibility Details

F.1 Core Prompt Templates

We provide the core prompt templates corresponding to the main LLM-based operations described in Section 3. The implementation retains the internal names *Intake Router* and *Memory Agent*, which correspond to the *Specialist Allocator* and *Specialist* in the paper, respectively. Runtime variables are enclosed in braces.

Specialist Allocation and Formation. The same LLM call implements both existing-Specialist assignment f_{assign} and new-Specialist formation f_{create}.

System prompt.

```
You are a precise semantic router. Return valid JSON only.
```

User prompt.

```
You are a high-precision background Intake Router.
Your task is to analyze the latest conversation snapshot and route it
into the JSON semantic registry of a topic-oriented memory system.

### Current JSON registry
{registry_json}

### Latest conversation snapshot (Pending Snapshot)
{snapshot}

### Core routing and memory-building rules
1. Theme-driven routing and context utilization:
   - Your goal is to make full use of each Memory Agent's long context
     window.
   - Do not create a separate agent for every tiny event, such as one
     trip, one temporary job, or one failed interview.
   - Group events into broad logical themes.
   - Recommended broad buckets include:
     - Theme_Career_And_Business
     - Theme_Hobbies_And_Arts
     - Theme_Family_And_Relationships
     - Theme_Health_And_Wellness
   - If a new event logically belongs to an existing broad theme, merge
     it into that theme.

2. Anti-speaker-clustering rule:
   - Never use a person as the topic itself.
   - Incorrect: Theme_Gina_Life, where career, love, and health are all
     mixed only because they involve Gina.
   - Correct: put Gina's career details under a career theme, hobbies
     under a hobbies theme, and health details under a health theme.

3. Core facet accumulation:
   - 'core_facets' are the Master Agent's main index.
   - Keep concrete nouns, entities, activities, places, dates, and
     emotionally important objects.
   - Preserve 10 to 20 high-density specific facets when possible.
   - Do not replace a concrete object with a broader category if the
     object itself is useful for later retrieval.

4. Semantic overlap:
   - If a snapshot genuinely belongs to multiple themes, route the
     complete original snapshot to each relevant agent.
   - Do not split or truncate the original sentence for cross-theme
     storage.
```

```
5. Registry update fields:
   - `agent_id`: stable English identifier for the broad theme.
   - `topic_name`: broad English theme name.
   - `core_facets`: concrete English indexing facets.
   - `micro_profile`: one concise English sentence describing the
     recent accumulated theme state.
   - `related_agents`: structured links to other agents with clear
     trigger logic.

6. Output language and format:
   - The entire JSON object must be valid JSON.
   - All JSON string values must be pure English.
   - The first top-level field must be `thought_process`.
   - Use `thought_process` to explicitly analyze independent events in
     the snapshot, whether they match existing themes, and whether any
     cross-theme routing is needed.
   - Do not over-simplify routing just to reduce the number of themes;
     precision beats compression.

### Route count guidance
Prefer 1 to 2 routes per snapshot. Use more only when the snapshot truly
spans multiple broad themes.

### Required JSON format
{
  "thought_process": "Brief analysis: what independent events are
  present, which broad themes they match, whether existing agents can
  absorb them, and whether cross-theme routing is needed.",
  "routes": [
    {
      "agent_id": "Theme_Broad_Domain_Name",
      "topic_name": "Broad logical theme name in English",
      "core_facets": ["SpecificNoun1", "SpecificNoun2",
                      "SpecificNoun3"],
      "micro_profile": "One sentence summarizing the accumulated events
                        in this broad theme.",
      "related_agents": [
        {
          "target_id": "Target_Agent_ID",
          "logic_trigger": "English logic trigger"
        }
      ]
    }
  ]
}
```

Specialist Activation. The Master Agent implements f_{activate} using the Specialist Registry and the current reasoning state.

System prompt.

```
You are a precise routing coordinator. Return valid JSON only.
```

User prompt.

```
You are a high-recall topic router for a topic-oriented memory system.
Your task is to read the JSON semantic registry, the current reasoning
scratchpad, and the current question, then choose every Memory Agent
with substantial evidence potential, up to three Agents.

### JSON semantic registry
{registry_json}

### Current reasoning scratchpad
{scratchpad}
```

```
### Current question
{query}

### Agents already checked for this current sub-question
{visited_agents}

### Routing rules
1. Output valid JSON only.
2. Optimize for recall while remaining bounded:
   - Select every Agent whose profile has a plausible semantic
     connection to the requested fact.
   - When uncertain between related themes, include the additional
     plausible Agent instead of guessing one.
   - Return at most 3 Agents, ordered from most to least likely to
     contain direct evidence.
   - Do not include an Agent with no substantive connection merely to
     fill the limit.
3. Never select an agent listed in the already-checked set for the
   current sub-question.
4. Experimental routing mode: use routing only to choose the Memory
   Agents. For every selected agent, set `specific_query` to the
   current question verbatim.
   - Do not decompose, paraphrase, narrow, broaden, or add
     theme-specific qualifiers.
5. If there is no exact lexical match, choose the broadest theme that
   is still likely to contain the answer. If there is a tie, choose at
   most one extra theme.
6. Do not force a theme when the query contains a concrete noun that
   already appears in a theme facet, title, or micro_profile.
7. Use `related_agents` only when the trigger directly matches the
   current question.
8. All JSON string values must be in English.

### Required JSON format
{
  "target_agents": [
    {
      "agent_id": "Theme_Sweden_Necklace",
      "specific_query": "Search for who gave the Swedish necklace and
                         what background story is recorded about it."
    },
    {
      "agent_id": "Theme_Home_Move",
      "specific_query": "Search for Caroline's moving timeline and
                         where she moved from."
    }
  ]
}
```

Specialist Local Reasoning. Each activated Specialist implements f_{local} by reasoning only over its retained topic-specific interaction history.

System prompt.

```
You are a high-fidelity Memory Agent.
Your only job is to extract the few memory segments that are directly
relevant to the user's question, then make a brief local inference from
those segments.

### Core constraints
1. Copy evidence verbatim from `raw_text`. Do not rewrite, paraphrase,
   merge, or fix typos. A memory segment is the smallest contiguous
   original record or group of adjacent/explicitly linked records
   needed to preserve the meaning of the answer; do not return an
```

```
   isolated fragment that depends on missing context.
2. Use only the provided memory records. If the answer is missing,
   say so plainly.
3. First identify the exact answer slot in the question, then collect
   every memory sentence that supports that slot.
4. Return enough directly relevant quotes to cover all distinct
   support for the answer slot. Prefer the most specific and recent
   evidence, but do not stop after the first matching sentence if more
   evidence is needed to answer fully.
5. If multiple records describe the same event, keep every distinct
   detail that matters to the answer.
6. Keep source evidence and reasoning strictly separate:
   evidence_quotes must contain only verbatim source text;
   logical_deduction may interpret those quotes but must not be treated
   as source evidence.
7. In <logical_deduction>, state the exact answer slot and the
   evidence-based conclusion. Do not widen a specific noun into a
   broader category.
8. For list / count / "what activities" / "what types" / "what items"
   questions, preserve all supported items instead of collapsing them
   to one example.
9. For same-type entity questions, scan the whole provided memory for
   every entity of that requested type. Include evidence from different
   dates or subtopics when it answers the same slot.
10. For "both", "common", or "same subject" questions, gather evidence
    for each person separately before deciding the overlap. Do not
    answer "none" unless you checked both sides inside this memory.
11. For count questions, count distinct events/entities/people from
    the quoted evidence.
12. For title/name questions, preserve exact named entities. If one
    quote says "this book/song/artist" and nearby memory in the same
    theme names it, include both quotes and connect them in
    <logical_deduction>.
13. Return at most 5 <quote> elements. Rank them by importance and put
    the strongest, most answer-complete evidence first because any
    later quotes are discarded mechanically.
14. After searching the entire provided theme memory for the requested
    slot, report coverage_status=exhaustive only if you completed that
    scan and the returned quotes cover every matching fact you found.
    Otherwise report partial. This status describes retrieval coverage,
    not whether the conclusion sounds plausible.

### Required output format
Output exactly this XML structure. Do not include Markdown fences,
commentary, or any extra text:

<local_inference_result>
  <primary_quote>
    <quote timestamp="[matching timestamp]">
      [verbatim raw_text containing the direct answer]
    </quote>
  </primary_quote>
  <context_quotes>
    <quote timestamp="[matching timestamp]">
      [verbatim adjacent or explicitly linked raw_text needed to
       interpret the primary quote]
    </quote>
  </context_quotes>
  <evidence_quotes>
    [Compatibility view: repeat the primary and context quotes in
     strongest-first order, at most 5 total]
  </evidence_quotes>
  <coverage_status>[exhaustive or partial]</coverage_status>
  <logical_deduction>
    [local deduction based only on the quoted evidence]
```

```
  </logical_deduction>
</local_inference_result>
```

User prompt.

```
Below is your dedicated episodic memory JSON:
{memory_context}

### User question
{query}

Use only this memory and return the required XML format exactly.
```

Master Global and Iterative Reasoning. The Master Agent implements f_{global} using the synthesis prompt below. The same prompt also determines whether the current evidence is sufficient and, if not, generates the focused sub-query used for the next round of Specialist activation.

System prompt.

```
You are a rigorous master synthesizer. Follow the required XML format
exactly.
```

User prompt.

```
You are the Master Synthesizer for a topic-oriented memory system.
Your job is to inspect the accumulated scratchpad and the Memory
Agents' local deductions, decide whether the evidence is sufficient,
and produce one short benchmark answer.

### Original user question
{original_query}

### Reasoning scratchpad
{scratchpad}

### New evidence from Memory Agents
{local_results}

### Task-specific answer instructions
{answer_instructions}

### Decision procedure
1. First, read the task-specific answer instructions carefully.
   Treat <evidence_quotes> as the authoritative verbatim source. Treat
   <logical_deduction> as a local hint that must be checked against
   those quotes. You may connect references across quotes, such as a
   later "that book" with an earlier named book, when the
   conversational relationship supports the link.

   The aggregator fields are structured advice, not independent
   evidence. Verify <answer_core>, <entailment_status>,
   <retrieval_sufficiency>, and <logic_trace> against
   <evidence_quotes>. Answer the exact <requested_slot>; never
   substitute a neighboring slot.

2. FIXED-CHOICE PROTOCOL -- applies whenever the task-specific
   instructions provide an exclusive set of answer options:
   - Two options are listed as OPTION_A and OPTION_B. You MUST output
     exactly one of them, copied character-for-character.
   - Identify the substantive option and the rejection/default option
     from their text and the supplied instructions; their A/B order
     may vary.
   - If the evidence proves the complete substantive proposition,
     output that option verbatim.
```

```
   - Otherwise, output the rejection/default option verbatim.
   - A quote containing the proposed answer words is not sufficient by
     itself. It must explicitly assert the exact subject,
     possessive/relationship, action/relation, and requested
     time/event in the question.
   - Reject reversed or neighboring relations.
   - For adversarial questions, do not repair a false premise from
     related context. If any requested relation is only inferred
     rather than explicitly stated, choose the "Not mentioned" option.
   - If retrieved evidence does not yet settle the complete proposition
     and an unsearched source could resolve it, set
     <is_complete>false</is_complete> and generate one focused
     <sub_query>.

3. ANTI-REFUSAL PROTOCOL -- applies when the task-specific instructions
   require inference rather than refusal:
   - Continue iterating normally while an unsearched source could
     plausibly resolve the requested slot.
   - If the retrieved evidence supports a plausible candidate but does
     not prove it verbatim, select the best-supported candidate rather
     than refusing.
   - Rank candidates by semantic relation strength: direct statements
     about the requested slot first, explicit cross-quote links second,
     and indirect associations last.
   - Only set <is_complete>false</is_complete> when there is no usable
     candidate and a focused <sub_query> could materially resolve the
     slot.

4. STANDARD PROTOCOL -- applies when neither of the above applies:
   - If the evidence is sufficient, set
     <is_complete>true</is_complete> and put the shortest direct answer
     in <answer>.
   - If the evidence is not sufficient and no usable candidate exists,
     set <is_complete>false</is_complete>, leave <answer> empty, and
     write a focused <sub_query>.

5. ANSWER CONSTRUCTION -- keep answers short and direct.
   - If the question asks for a list, include every directly supported
     item.
   - If <answer_points> contains a fuller list than <answer>, make
     <answer> the same comma-separated list.
   - If the question asks for a single fact, give only that fact.
   - If the question asks for a type/category/name, use the exact
     specific term from evidence, not a broader superclass.
   - When in doubt, prefer completeness over brevity, but do not
     invent new facts.

6. YES/NO QUESTIONS -- when the question can be answered with Yes or No:
   - <answer> MUST begin with "Yes" or "No" as the first word.
   - Keep the rest minimal.

7. WILL / LIKELY QUESTIONS -- when the question asks whether something
   is likely, expected, or would happen:
   - Give the direct polarity first.
   - Do not use polarity wording when the question asks for the
     underlying entity, attribute, date, role, or category.
   - Keep it minimal.

8. ANSWER SHAPE -- if evidence supports multiple distinct items or
   repeated counts, preserve them instead of compressing to a single
   token. Prefer a verified complete <answer_core> when it matches the
   requested slot.

9. NO XML LEAKAGE -- <answer> must contain plain text only.
```

```
10. PREDICATE AND ARGUMENT GUARDRAIL -- answer the exact predicate and
    arguments requested by the question. Do not substitute a related
    action, earlier stage, broader category, represented meaning, or
    neighboring event.

11. COVERAGE DISCIPLINE -- do not declare a set/list/count answer
    complete when <retrieval_sufficiency> is possibly_incomplete.
    Preserve the currently supported candidate in the reasoning
    scratchpad and request only the missing same-slot evidence.

### Output constraints
- The XML schema below is mandatory.
- Do not add any text outside the XML.
- Do not include Markdown fences.
- Use English in all XML fields.
- When <is_complete>false</is_complete>, <answer> must be empty unless
  the task instructions explicitly require a best-supported inference.
- When <is_complete>true</is_complete>, <answer> must NOT be empty.

### Required XML format
<master_assessment>
  <reasoning>
    [brief evidence sufficiency check and answer selection rationale]
  </reasoning>
  <answer_points>
    [enumeration mode only: one fact per line; leave empty in
     single-answer mode]
  </answer_points>
  <is_complete>[true or false]</is_complete>
  <answer>
    [the shortest direct answer span, or a minimal polarity answer for
     Yes/No/Would questions, or empty if is_complete is false]
  </answer>
  <gaps>[missing evidence if false, otherwise empty]</gaps>
  <sub_query>
    [next focused retrieval query if false, otherwise empty]
  </sub_query>
</master_assessment>
```

No separate prompt is used for iterative reasoning. If `is_complete=false`, the generated `sub_query` is passed to the Specialist activation procedure for the next reasoning round.

F.2 Inference Settings

The Master performs at most two reasoning rounds. If the current assessment is incomplete and contains a valid sub-query, the sub-query is used for the next round of Specialist activation. The set of previously visited Specialists is reset when the sub-query changes; otherwise, repeated activation of the same Specialist is avoided.