GAM: HIERARCHICAL GRAPH MEMORY FOR LLM-BASED AGENTS

Zhaofen Wu[1*] **Hanrong Zhang**[2*†] **Fulin Lin**[1] **Wujiang Xu**[3] **Xinran Xu**[1]
Yankai Chen[4,5] **Shaowen Chen**[1] **Henry Peng Zou**[2] **Weizhi Zhang**[2] **Xue Liu**[4,5]
Philip S. Yu[2] **Hongwei Wang**[1†]
[1]Zhejiang University [2]University of Illinois at Chicago [3]Rutgers University [4]MBZUAI
[5]McGill University
{zhaofen1.25, hongweiwang}@intl.zju.edu.cn, {hzhan135, psyu}@uic.edu

ABSTRACT

To sustain coherent long-term interactions, Large Language Model (LLM) agents must navigate the tension between acquiring new information and retaining prior knowledge. Current unified stream-based memory systems facilitate context updates but remain vulnerable to interference from transient noise. Conversely, discrete structured memory architectures provide robust knowledge retention but often struggle to adapt to fluid narrative evolution. To address this, we propose **GAM**, a hierarchical **G**raph-based **A**gentic **M**emory framework that explicitly decouples memory encoding from consolidation to effectively resolve the conflict between rapid context perception and stable knowledge retention. By isolating ongoing dialogue in an event progression graph and integrating it into a topic associative network only upon semantic shifts, our approach minimizes interference while preserving long-term consistency. Additionally, we introduce a graph-guided, multi-factor retrieval strategy to enhance context precision. Experiments on LoCoMo and LongDialQA indicate that our method consistently outperforms state-of-the-art baselines in both reasoning accuracy and efficiency.

1 INTRODUCTION

Large Language Model (LLM) agents have demonstrated remarkable capabilities in natural language understanding, long-horizon complex reasoning, and question answering (Didolkar et al., 2024; Anwar et al., 2025; Ma et al., 2025). However, enabling these agents to maintain coherent long-term interactions remains a significant challenge (Du et al., 2025; Yang et al., 2025a). A robust agentic memory must rapidly capture real-time interactions while safeguarding established knowledge from transient noise corruption (Tan et al., 2025).

Existing architectures often struggle to resolve this conflict. Recent approaches typically maintain a continuous, linear flow of information to handle ongoing interactions (Zhong et al., 2024; Lee et al., 2024; Packer et al., 2024; Kang et al., 2025). We categorize these paradigms as Unified Stream-based Memory Systems. As depicted in the left part of Figure 1, these methods perform updates directly on the active memory stream. This continuous exposure introduces a high risk of Memory Contamination (Chen et al., 2025). Specifically, this continuous exposure leads to Memory Loss (Jia et al., 2025). In this phenomenon, established nodes become structurally isolated and forgotten due to weak connectivity. Furthermore, it causes Semantic Drift (Yang et al., 2025b). This occurs when distinct topics are erroneously conflated, thereby distorting the agent's thematic consistency. Conversely, Discrete Structured Memory Architectures ensure stability by organizing information into rigid schemas (Zhang et al., 2025; Edge et al., 2024). However, they often lack the agility to update narrative flows in real-time. This rigidity results in fragmented discourse representations. Consequently, they fail to capture the continuous evolution of long-term agent-user dialogues. While some recent studies attempt to address structural dynamism, they often rely on artificial discrete state changes or focus solely on retrieval optimization (Anokhin et al., 2024; Rasmussen et al.,

[*]Equal contribution.
[†]Corresponding authors.

2025). These methods fail to fundamentally resolve the conflict between immediate encoding and long-term consolidation. In contrast, our approach explicitly decouples these processes to ensure robustness.

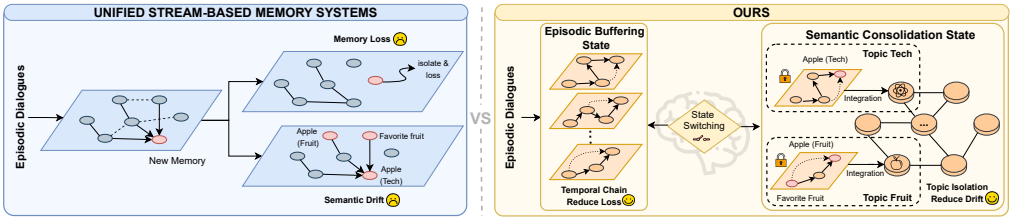

Figure 1: Comparison between Unified Stream-based Memory Systems (left) and our framework (right). While unified architectures suffer from Memory Loss and Semantic Drift due to direct updates, our solution employs a state-switching mechanism: the Episodic Buffering State maintains temporal chains to reduce loss, while the Semantic Consolidation State ensures topic isolation to integrate knowledge and reduce drift.

To address these challenges, we propose the Graph-based Agentic Memory (GAM) framework, which uses a semantic-event-triggered mechanism to decouple encoding from consolidation. Drawing inspiration from sleep-dependent memory consolidation (Chang et al., 2025), our architecture separates the memory lifecycle into two phases, as illustrated in the right part of Figure 1. An Episodic Buffering Phase constructs a local graph to capture real-time dependencies and strictly isolate transient context, while an event-driven Semantic Consolidation Phase integrates these details into the global network only upon semantic shifts. This ensures that global memory is updated only with semantically complete units, unlike prior stream-based methods. Finally, a graph-guided multi-factor retrieval paradigm leverages structured cross-layer links to empower the agent to recall precise context.

The main contributions of our work are as follows:

- We propose the **GAM** framework, underpinned by a **Hierarchical Graph Memory Architecture**, which structurally mitigates the conflict between rapid context perception and stable knowledge retention. By physically separating the global Topic Associative Network from local Event Progression Graphs, we ensure robust write isolation for narrative buffering.
- We design a **State-Based Episodic-Semantic Consolidation** mechanism that dynamically transitions the system between an Episodic Buffering State and a Semantic Consolidation State. This state-switching strategy replaces arbitrary triggers with semantic divergence detection, ensuring that memory updates occur only at semantically complete boundaries to minimize contamination.
- We introduce a **Graph-Guided Multi-Factor Retrieval** strategy that effectively bridges the decoupled storage layers. By integrating temporal, confidence, and role-based signals into a top-down traversal, this mechanism allows the agent to recall precise episodic details grounded in high-level semantic themes.
- We demonstrate through extensive evaluations on LoCoMo and LongDialQA benchmarks that our approach significantly outperforms state-of-the-art baselines in both reasoning accuracy and computational efficiency.

2 Methodology

2.1 Problem Formulation and Overview

We formalize memory management as an inference-time online decision process that balances immediate accessibility against long-term interference. Let a dialogue sequence $D = \{u_1, \ldots, u_t\}$ be

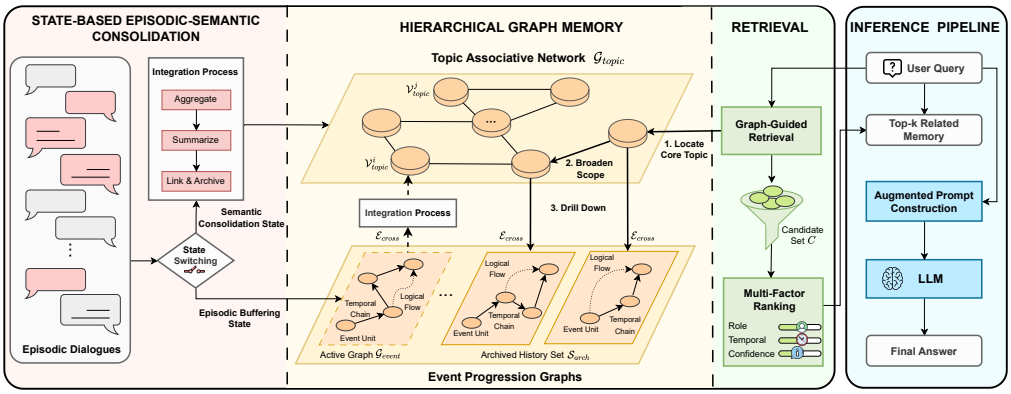

Figure 2: The architecture of GAM. The leftmost State-Based Consolidation buffers dialogue in the Active Graph $\mathcal{G}_{event}$ until topic shifts trigger the Integration Process. The central Hierarchical Graph Memory connects the global Topic Associative Network $\mathcal{G}_{topic}$ with the grounded Archived History Set $\mathcal{S}_{arch}$. The rightmost Graph-Guided Retrieval locates and expands semantic anchors, then drills down into archives to extract and rank event details via multi-factor signals.

maintained by a hierarchical memory graph $\mathcal{H}_t$. Rather than optimizing model parameters, GAM seeks a policy that minimizes the cumulative operational cost of updating memory while preserving retrieval quality:

$$\pi^* = \arg\min_{\pi} \mathbb{E}\left[\sum_{t=0}^{T}\left(C_{\text{enc}}(u_t, \mathcal{H}_t) + C_{\text{inter}}(\mathcal{G}_{\text{topic}}^{(t)}, \mathcal{G}_{\text{event}}^{(t)})\right)\right], \tag{1}$$

where C_{enc} denotes the operational cost of incorporating new utterances into the active memory state, and C_{inter} denotes the expected retrieval degradation caused by letting transient local content interfere with the stable global store. In the concrete architecture introduced below, the local role is instantiated by the Event Progression Graph $\mathcal{G}_{\text{event}}$, while the long-term global role is instantiated by the Topic Associative Network $\mathcal{G}_{\text{topic}}$.

Because evaluating future interference exactly is intractable online, we approximate the policy in Equation 1 through a semantic divergence indicator $b_t \in \{0, 1\}$ that signals when the accumulated local context has drifted far enough from the global state to justify consolidation:

$$b_t = \mathbb{I}(\Delta(\mathcal{G}_{\text{event}}^{(t)}, \mathcal{G}_{\text{topic}}^{(t)}) > \epsilon), \tag{2}$$

where $b_t = 1$ signifies a structural shift requiring consolidation. Our objective is therefore not to solve Equation 1 exactly, but to use b_t as a low-cost policy trigger that reduces the long-run interference term without sacrificing fast online updates.

To address this optimization conflict, we propose the GAM framework. This framework employs a state-based consolidation mechanism within a hierarchical graph architecture to structurally separate the memory life cycle. As illustrated in Figure 2, our system satisfies the aforementioned objectives through three coordinated stages: (1) Episodic Buffering Phase. To optimize perception, we isolate ongoing dialogue in a local Event Progression Graph $\mathcal{G}_{\text{event}}$ whose structure is defined in Section 2.2. This shielding mechanism protects the global store from temporary noise while the system operates in the Episodic Buffering State described in Section 2.3. (2) Semantic Consolidation Phase. Upon detecting a semantic indicator b_t, the system triggers an atomic update to ensure stability. As detailed in Section 2.3, buffered data is summarized and merged into the global Topic Associative Network $\mathcal{G}_{\text{topic}}$ only when narrative units are semantically complete. (3) Graph-Guided Retrieval. To bridge these separated layers during inference, we employ the multi-factor traversal strategy elaborated in Section 2.4. This process integrates temporal signals and confidence scores to achieve precise context recovery from both local details and global themes. The detailed end-to-end execution flow of GAM is provided in Algorithm 1 in Appendix B.

2.2 HIERARCHICAL GRAPH MEMORY ARCHITECTURE

To implement the structural separation described in Section 2.1, we define the time-varying memory structure $\mathcal{H}_t$ as a composite graph. This architecture separates the storage of consolidated knowledge from the temporary context buffer and the historical archives. We formally define it as:

$$\mathcal{H}_t = \{\mathcal{G}_{\text{topic}}^{(t)}, \mathcal{G}_{\text{event}}^{(t)}, \mathcal{S}_{\text{arch}}^{(t)}, \mathcal{E}_{\text{cross}}^{(t)}\}, \tag{3}$$

where $\mathcal{S}_{\text{arch}}^{(t)}$ represents the set of archived event graphs that have been consolidated, while $\mathcal{G}_{\text{event}}^{(t)}$ denotes the active buffer. This decomposition makes the abstract roles from Equation 1 explicit: $\mathcal{G}_{\text{event}}$ realizes the transient local state used for rapid perception, $\mathcal{G}_{\text{topic}}$ realizes the stable global substrate, and $\mathcal{S}_{\text{arch}}$ with $\mathcal{E}_{\text{cross}}$ preserve grounded evidence for later retrieval. We omit the time subscript t when describing static structures for brevity.

The consolidated knowledge is represented as the Topic Associative Network which serves as the foundation for long-term retention. Its structure is defined as:

$$\mathcal{G}_{\text{topic}} = (\mathcal{V}_{\text{topic}}, \mathcal{E}_{\text{topic}}). \tag{4}$$

In this layer, nodes $\mathcal{V}_{\text{topic}}$ represent high-level semantic clusters or abstract themes derived from historical interactions. The edges $\mathcal{E}_{\text{topic}}$ capture the deep semantic correlations between them. To capture subtle dependencies that heuristic metrics might miss, we quantify these edge weights via a semantic scorer based on large language models. This process utilizes the prompt detailed in Appendix F.1. While computationally intensive, this precise calculation is executed exclusively during the consolidation phase. This timing ensures that the structural precision of $\mathcal{G}_{\text{topic}}$ is achieved without compromising the interaction capabilities of the system.

In parallel, the temporary context is maintained in the active Event Progression Graph which is optimized for rapid updates. We define this structure as:

$$\mathcal{G}_{\text{event}} = (\mathcal{V}_{\text{event}}, \mathcal{E}_{\text{event}}), \tag{5}$$

where nodes $\mathcal{V}_{\text{event}}$ correspond to atomic interaction units such as user utterances or system responses which are captured in real-time. The edges $\mathcal{E}_{\text{event}}$ model the temporal and causal evolution of the current dialogue flow. Unlike the global layer, this structure is designed for rapid append operations which minimizes the encoding latency $\mathcal{L}_{\text{enc}}$.

Finally, we establish cross-layer associations $\mathcal{E}_{\text{cross}}$ to bridge the gap between episodic details and semantic themes. These associations act as an index that connects topic nodes in the global graph to the archived snapshots of event graphs within $\mathcal{S}_{\text{arch}}$. This structure allows the retrieval mechanism to access the exact historical evidence associated with a theme without relying on the volatile active buffer.

2.3 STATE-BASED EPISODIC-SEMANTIC CONSOLIDATION

Our system models narrative dynamism as a finite-state machine that transitions between two distinct states based on semantic progression.

Implementation of Semantic Boundary Detection. To realize the theoretical boundary condition defined in Equation 2, we require a mechanism to detect precisely when the semantic divergence exceeds the tolerance ϵ. Since explicitly computing the high-dimensional distance Δ is computationally expensive and sensitive to lexical noise, we implement the detection function as a semantic discrimination task driven by an LLM $\mathcal{M}_\theta$.

In this framework, the LLM serves as a neural proxy for the thresholding function. The system maintains a fixed-capacity episodic buffer B_t with a limit of 2048 tokens, derived from the linearized content of the current Event Progression Graph $\mathcal{G}_{\text{event}}$. Crucially, the discriminator is not queried at every turn. Instead, it is triggered only by sparse maintenance events such as session-end markers, natural interaction pauses, or buffer overflow. This process uses the structural instruction prompt defined in Appendix F.3. In practice, ϵ is therefore not instantiated as a rigid numeric graph-distance threshold; the instruction strictness of the semantic discrimination prompt acts as a heuristic proxy for ϵ. A positive detection where b_t equals 1 serves as an indicator that the semantic distance has

breached the stability threshold and triggers a transition to the Semantic Consolidation State. If no sparse trigger is observed before the buffer reaches capacity, overflow forces a consolidation check, and semantically related split segments can later be reconnected through strong topic-level edges.

Episodic Buffering State. In this state, the system focuses on capturing details of the ongoing dialogue. Incoming utterances are constructed as atomic event units e_t and appended to the local graph by updating its node and edge sets:

$$\mathcal{V}_{\text{event}}^{(t)} \leftarrow \mathcal{V}_{\text{event}}^{(t-1)} \cup \{e_t\}, \quad \mathcal{E}_{\text{event}}^{(t)} \leftarrow \mathcal{E}_{\text{event}}^{(t-1)} \cup \mathcal{E}_{\text{temp}}^{(t)}, \tag{6}$$

where $\mathcal{E}_{\text{temp}}^{(t)}$ denotes the sequential edges that link the new event to the immediate context. This state functions as a strict write isolation buffer. By confining high-frequency updates to this local scope, we protect the global Topic Associative Network $\mathcal{G}_{\text{topic}}$ from temporary noise. This mechanism effectively separates the acquisition of new information from the modification of long-term memory. This separation mitigates the contamination risk outlined in Equation 2.

Semantic Consolidation State. Upon detecting a topic boundary where b_t equals 1, the system transitions to this state to consolidate the completed narrative unit. This process executes a graph merging operation that transforms the buffered subgraph $\mathcal{G}_{\text{event}}$ into a consolidated semantic node v_{new}. To address the trade-off between abstract reasoning and detailed recall, we design v_{new} with a dual-granularity representation:

$$v_{\text{new}} = \{c_{\text{sum}}, c_{\text{raw}}\}, \tag{7}$$

where c_{sum} is generated by prompting an LLM with the instruction detailed in Appendix F.2 to summarize the semantic content of the event graph. This summary enables high-level thematic reasoning. Simultaneously, c_{raw} is constructed by concatenating the raw textual content of all event nodes within the buffer. This component preserves fine-grained details and prevents information loss during summarization.

Subsequently, the system integrates v_{new} into the global network $\mathcal{G}_{\text{topic}}$ by establishing semantic edges. To avoid expensive full-graph reasoning over every topic node, we adopt a coarse-to-fine candidate selection strategy. The summary c_{sum} first retrieves the top-5 nearest topic nodes using vector similarity, and only this compact candidate set is passed to the LLM-based semantic scorer introduced in Section 2.2. Utilizing the instruction prompt detailed in Appendix F.1, we pair the summary of v_{new} with each candidate topic node to query the model for a specific relationship type and a confidence score. New edges $\mathcal{E}_{\text{new}}$ are established when this confidence score exceeds a semantic threshold τ. This coarse-to-fine procedure keeps the logical precision of LLM-based relation modeling while avoiding $O(N)$ graph-wide scoring. The global topology is updated as:

$$\mathcal{V}_{\text{topic}}^{(t)} \leftarrow \mathcal{V}_{\text{topic}}^{(t-1)} \cup \{v_{\text{new}}\}, \quad \mathcal{E}_{\text{topic}}^{(t)} \leftarrow \mathcal{E}_{\text{topic}}^{(t-1)} \cup \mathcal{E}_{\text{new}}. \tag{8}$$

Once integration is complete, the current event graph structure is archived into the set $\mathcal{S}_{\text{arch}}^{(t)}$ and permanently linked to v_{new} as grounded evidence through the creation of cross-layer edges $\mathcal{E}_{\text{cross}}$. The system then resets the active local buffer $\mathcal{G}_{\text{event}}$ to an empty state and reverts to the Episodic Buffering State for the subsequent narrative unit.

2.4 Graph-Guided Multi-Factor Retrieval

While the dual-phase architecture explicitly separates storage to promote stability, effective interaction requires a unified view of the memory system. We propose a graph-guided retrieval mechanism that acts as a logical bridge to traverse the hierarchical structure. This approach aims to retrieve context that is both semantically deep and temporally precise by integrating signals from the separated storage layers.

Unlike flat vector retrieval, our process exploits the topological structure of $\mathcal{H}_t$ in a top-down, expand-and-drill manner. The retrieval process operates in three stages:

Semantic Anchoring and Expansion. First, the system identifies the most relevant semantic anchors $\mathcal{V}_{\text{top}}$ in the Topic Associative Network $\mathcal{G}_{\text{topic}}$ via vector similarity. To capture latent dependencies beyond direct lexical matching, we leverage the global graph structure to expand the search

scope. Specifically, we include the first-order neighbors of the top-k nodes, utilizing the semantic edges $\mathcal{E}_{\text{topic}}$ to bring in contextually related themes. The final set of semantic anchors $\mathcal{V}_{\text{anchor}}$ is defined as:

$$\mathcal{V}_{\text{anchor}} = \mathcal{V}_{\text{top}} \cup \{v \in \mathcal{V}_{\text{topic}} \mid \exists u \in \mathcal{V}_{\text{top}}, (u, v) \in \mathcal{E}_{\text{topic}}\}. \tag{9}$$

Structural Drill-Down. From these expanded anchors, the system traverses the cross-layer links $\mathcal{E}_{\text{cross}}$ to access specific archived Event Progression Graphs stored in the archive set $\mathcal{S}_{\text{arch}}$. These archived graphs constitute the raw source of the topic nodes. Formally, we define the candidate set $\mathcal{C}$ by aggregating all event nodes from the archival graphs linked to any node in the anchor set:

$$\mathcal{C} = \bigcup_{u \in \mathcal{V}_{\text{anchor}}} \{v \in \mathcal{V}(\mathcal{G}') \mid \mathcal{G}' \in \mathcal{S}_{\text{arch}} \wedge (u, \mathcal{G}') \in \mathcal{E}_{\text{cross}}\}, \tag{10}$$

where $\mathcal{V}(\mathcal{G}')$ denotes the node set of an archived graph $\mathcal{G}'$. This strategy ensures that the agent accesses precise episodic details strictly aligned with both the direct and latent macro-level themes required by the query.

Multi-Factor Re-ranking. To rank the diverse candidates in $\mathcal{C}$, we employ a re-ranking strategy that combines deep semantic reasoning with explicit contextual signals. Given a user query q, we first compute a base semantic probability $P_{\text{sem}}(v|q)$ for each memory unit v using a cross-encoder model. This model captures subtle semantic dependencies that bi-encoders might miss. To further enforce consistency with query constraints, we apply a multiplicative signal modulation mechanism defined below:

$$\text{Score}(v, q) = P_{\text{sem}}(v|q) \cdot \prod_{k \in \mathcal{K}} \beta_k^{\mathbb{I}_k(v,q)}, \tag{11}$$

where $\mathcal{K} = \{\text{time}, \text{conf}, \text{role}\}$ indexes the set of modulation factors. The base probability is modulated by boosting factors $\beta_k > 1$ which are activated by indicator functions $\mathbb{I}_k(v, q) \in \{0, 1\}$. Specifically, we incorporate three pragmatic constraints to refine the retrieval scope. First, a temporal factor β_{time} activates when the memory contains temporal expressions relevant to a time-sensitive query which promotes chronological precision. Second, we include an intrinsic confidence factor β_{conf} which prioritizes information that successfully passed the self-consistency verification during its encoding phase. Finally, to address the complexity of multi-party interactions, we introduce a role-centric contextualization factor β_{role}. This factor helps disentangle mixed narrative threads by verifying if the memory source aligns with the target interlocutors implied by the query.

This multiplicative formulation ensures that candidates that simply match keywords without semantic relevance are not falsely promoted as their base semantic probability would remain low. We conducted a sensitivity analysis on these boosting factors β ranging from 1.0 to 2.0 to evaluate system robustness. As detailed in Appendix E.1, our framework maintains consistent performance within a wide hyperparameter margin. This result suggests that the integration of structured priors contributes to retrieval quality alongside semantic matching.

3 Experiments

3.1 Experimental Settings

Datasets. We evaluate on two benchmarks for long-term interaction. LoCoMo (Maharana et al., 2024) features long open-domain dialogues across five reasoning categories. Following MemoryOS (Kang et al., 2025) and Mem0 (Chhikara et al., 2025), we focus on the first four categories to assess stability. LongDialQA (Kim et al., 2024) simulates multi-party interactions to strictly assess context retention under adversarial settings.

Baselines. We compare diverse architectures across three representative paradigms. First, heuristic approaches like MemoryBank (Zhong et al., 2024) and ReadAgent (Lee et al., 2024) rely on passive forgetting curves or gist compression. Second, we benchmark unified stream-based systems, including OS-inspired models such as MemGPT (Packer et al., 2024) and MemoryOS (Kang et al., 2025), alongside the industrial standard Mem0 (Chhikara et al., 2025). Finally, we include the self-evolving agent A-Mem (Xu et al., 2025) to assess recent self-reflective update mechanisms.

Table 1: Performance comparison on the LoCoMo dataset. We evaluate our method against various baselines across four different LLM backbones. **F1** and **B-1** denote F1 Score and BLEU-1 Score, respectively. The best results for each backbone are highlighted in bold, and the second best are underlined.

Model	Method	Multi-Hop		Temporal		Open-Domain		Single-Hop		Avg	
		F1	B-1	F1	B-1	F1	B-1	F1	B-1	F1	B-1
Llama 3.2-3B	ReadAgent	2.05	1.96	4.33	4.33	7.53	6.39	3.52	2.75	4.36	3.86
	Memorybank	7.35	5.34	3.90	3.41	4.91	5.42	8.76	7.04	6.23	5.30
	Memgpt	7.59	5.09	3.25	3.15	5.74	5.55	7.91	6.64	6.12	5.11
	A-Mem	17.44	11.74	26.38	19.50	12.53	11.83	28.14	23.87	21.12	16.73
	MemoryOS	21.03	13.75	19.38	13.73	11.46	8.58	28.25	20.03	20.03	14.02
	Mem0	27.51	18.51	30.08	20.73	**21.33**	15.00	41.52	30.89	30.11	21.28
	Ours	**28.74**	**22.35**	**46.12**	**37.24**	21.18	**18.33**	**49.02**	**42.18**	**36.27**	**30.02**
Qwen 2.5-7B	ReadAgent	5.37	4.02	2.82	2.74	5.86	4.78	5.23	4.13	4.82	3.92
	Memorybank	6.22	5.39	2.58	2.27	8.53	7.58	6.17	4.87	5.88	5.03
	Memgpt	6.17	4.97	2.18	2.77	6.33	5.93	8.95	7.33	5.91	5.25
	A-Mem	21.85	13.40	27.54	22.49	14.16	13.06	33.25	29.00	24.20	19.49
	MemoryOS	29.37	21.40	28.96	20.69	20.22	16.12	36.89	30.81	28.86	22.25
	Mem0	32.86	23.99	41.22	32.22	18.24	15.63	49.21	42.83	35.38	28.67
	Ours	**35.32**	**25.77**	**48.97**	**39.68**	**21.14**	**17.89**	**54.58**	**48.62**	**40.00**	**32.99**
Qwen 2.5-14B	ReadAgent	4.33	2.99	2.66	2.71	4.93	5.53	5.16	4.05	4.27	3.82
	Memorybank	3.65	3.16	2.59	2.33	7.33	7.45	4.71	3.72	4.57	4.17
	Memgpt	8.37	5.42	4.05	3.64	11.81	11.52	11.70	9.79	8.98	7.59
	A-Mem	26.55	16.56	28.30	23.02	16.74	14.89	36.95	32.58	27.14	21.76
	MemoryOS	27.46	19.14	23.38	16.80	19.27	14.96	26.31	21.92	24.11	18.20
	Mem0	31.45	24.59	50.20	**39.85**	17.66	14.43	49.85	43.78	37.29	30.66
	Ours	**33.32**	**25.38**	**50.25**	39.33	**20.40**	**18.50**	**57.55**	**50.99**	**40.38**	**33.55**
GPT-4o-mini	ReadAgent	9.15	6.48	12.60	8.87	5.31	5.12	9.67	7.66	9.18	7.03
	Memorybank	5.00	4.77	9.68	6.99	5.56	5.94	6.61	5.16	6.71	5.71
	Memgpt	26.65	17.72	25.52	19.44	9.15	7.44	41.04	34.34	25.59	19.73
	A-Mem	27.02	20.09	45.85	36.67	12.14	12.00	44.65	37.06	32.41	26.45
	MemoryOS	**35.27**	**25.22**	41.15	30.76	20.02	16.52	48.62	42.99	36.27	28.87
	Mem0	32.78	20.87	**56.42**	**45.89**	20.45	15.82	51.81	43.23	40.37	31.45
	Ours	**35.88**	**27.96**	51.96	42.26	**28.12**	**24.51**	**56.58**	**51.18**	**43.14**	**36.48**

Metrics and Implementation. We employ F1 Score for entity capture and BLEU-1 for lexical accuracy. Implemented via PyTorch and HuggingFace, we leverage Ollama and LiteLLM for efficient inference. We evaluate performance across models of varying scales, including Llama-3.2-3B-Instruct, Qwen2.5-7B-Instruct, Qwen2.5-14B-Instruct and GPT-4o-mini. All backbone LLMs are used as off-the-shelf instruct-tuned models, and GAM itself is a training-free memory framework without any additional fine-tuning of backbone parameters. The architecture uses all-MiniLM-L6-v2 sentence embeddings for indexing and cross-encoder/ms-marco-MiniLM-L-6-v2 for re-ranking. We set the retrieval size k to 10 based on the sensitivity analysis detailed in Appendix E.2. Modulation factors are set to $\beta_{\text{time}} = 1.4$, $\beta_{\text{role}} = 1.4$, and $\beta_{\text{conf}} = 1.2$, chosen from the stable regions in Appendix E.1 rather than aggressive per-dataset tuning. In implementation, the role indicator is triggered by lightweight speaker matching, while temporal and confidence cues are inferred with lightweight prompts. All experiments were conducted on NVIDIA RTX 4090 GPUs.

3.2 Main Results

Performance on LoCoMo. Table 1 presents the comparative results across four LLM backbones where our framework achieves the best average performance in most settings. Our method is particularly effective in complex reasoning tasks. For instance, with the Qwen 2.5-7B backbone, we surpass the strong commercial baseline Mem0 (Chhikara et al., 2025) by over 18% in Temporal F1 score. Even on the larger GPT-4o-mini model, where baselines perform strongly, our approach maintains

Table 2: Performance comparison on the LongDialQA dataset (Average scores). **F1** and **B-1** denote F1 Score and BLEU-1 Score, respectively. The best results for each backbone are highlighted in bold, and the second best are underlined.

Method	Llama 3.2-3B		Qwen 2.5-7B		Qwen 2.5-14B		GPT-4o-mini	
	F1	B-1	F1	B-1	F1	B-1	F1	B-1
ReadAgent	3.97	4.04	6.41	6.35	8.71	8.85	4.50	3.16
Memorybank	3.93	3.60	7.22	7.05	9.86	10.24	5.95	6.06
Memgpt	3.90	3.36	4.94	4.43	5.83	5.81	3.87	3.28
A-Mem	4.11	3.27	5.49	5.79	8.53	7.93	6.49	5.97
MemoryOS	5.85	4.46	6.76	5.11	9.07	8.39	10.61	9.31
Mem0	<u>6.85</u>	<u>5.75</u>	<u>10.27</u>	<u>9.91</u>	<u>11.48</u>	<u>10.94</u>	<u>10.90</u>	<u>9.84</u>
Ours	**7.36**	**6.85**	**12.55**	**12.43**	**11.86**	**11.66**	**11.18**	**10.14**

Table 3: Comparison of resource consumption and performance on the LoCoMo dataset.

Method	Tokens/Q	Time (s)	F1
A-Mem	4221.12	2.21	24.20
MemoryOS	3400.41	154.22	28.86
Mem0	1533.94	**0.51**	35.38
Ours	**1370.18**	0.80	**40.00**

the highest Average F1 score of 43.14, which demonstrates superior global consistency. These gains suggest that our dual-phase architecture provides a more robust structural foundation for long-term memory than purely retrieval-optimized systems. We note two informative exceptions. First, Mem0 remains stronger on GPT-4o-mini for Temporal F1 (56.42 versus 51.96), suggesting that when the backbone already has strong temporal reasoning and large context utilization, direct retrieval of raw traces can preserve exact chronological cues that GAM may partially compress during consolidation. Second, Open-Domain is consistently the most difficult category for GAM because these questions are weakly localized and less constrained by temporal or role cues, making broad thematic coverage more important than the structured pruning that benefits focused reasoning tasks.

Performance on LongDialQA. As shown in Table 2, GAM consistently outperforms all baselines in complex multi-party narratives. Most notably, on the smaller Qwen 2.5-7B model, it achieves an Average F1 of 12.55, surpassing MemoryOS(Kang et al., 2025) by 86%. This suggests that our structured memory effectively compensates for the limited context capacity of smaller models. Further analysis in Appendix C.1 confirms this advantage holds across all sub-datasets, indicating that our Semantic-Event-Triggered mechanism successfully mitigates interference from frequent speaker switching.

Efficiency Analysis. We evaluate the computational efficiency of different memory frameworks by analyzing the average token consumption and inference latency. As shown in Table 3, our method achieves the lowest token consumption (1,370/query), reducing costs by 11% compared to Mem0. While Mem0 shows slightly lower latency, our approach secures a superior trade-off by delivering a 13% F1 gain at comparable speeds, while outperforming A-Mem and MemoryOS in both metrics.

3.3 Ablation Study

To validate the contribution of each component, we performed an ablation study on the LoCoMo dataset using Qwen2.5-7B by systematically removing the Topic Associative Network (w/o TAN), Event Progression Graph (w/o EPG), Semantic-Event-Triggered State Switching Mechanism (w/o SSM), and Multi-Factor Retrieval (w/o MFR).

The average results presented in Table 4 show that our full method outperforms all variants. Detailed performance breakdowns across all task categories are provided in Appendix C.2. The w/o EPG variant suffers from the most substantial degradation, suggesting that the narrative structure is fundamental for chronological robustness. The drop in w/o SSM underscores the necessity of decoupling encoding from consolidation to prevent contamination. Finally, declines in w/o TAN

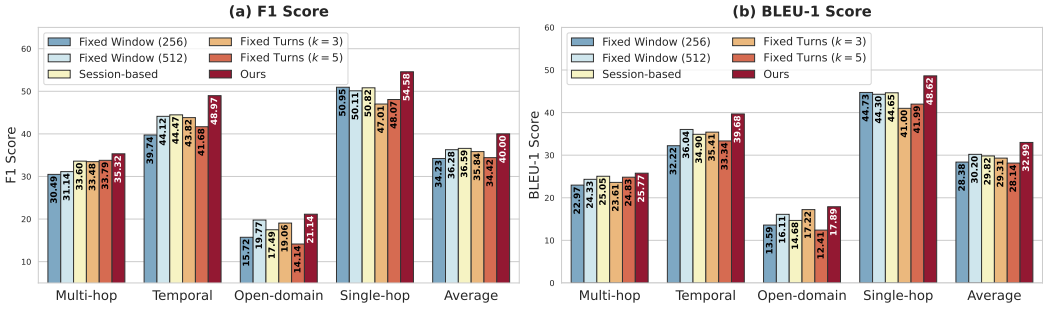

Figure 3: **Impact of Partitioning Strategies on LoCoMo.** Our semantic-boundary approach outperforms heuristic methods (Fixed Window, Fixed Turns, Session-based) across all metrics.

and w/o MFR validate that high-level topic organization and multi-signal fusion are essential for cross-session reasoning and precise retrieval.

Table 4: Ablation study summary (Average F1 and BLEU-1). Full results across all reasoning categories are detailed in Table 6 in Appendix C.2.

Method	**Avg**	
	F1	**B-1**
w/o TAN	35.07	29.00
w/o MFR	35.94	29.28
w/o SSM	32.58	26.13
w/o EPG	25.06	20.76
Ours	**40.00**	**32.99**

3.4 Topic Partitioning Strategies Analysis

To validate the necessity of aligning memory consolidation with semantically complete boundaries, we compare our semantic-event-triggered mechanism against three heuristic strategies: (1) **Fixed Window** (256 / 512 tokens), (2) **Fixed Turns** ($k \in \{3, 5\}$), and (3) **Session-based** partitioning. As shown in Figure 3, our method consistently outperforms all baselines, achieving the highest average F1 score of 40.00. Notably, the *Fixed Window (256)* strategy suffers the lowest performance of 34.23. This confirms that arbitrary hard cuts lead to context fragmentation. Such cuts sever the Logical Flow edges essential for reasoning. This advantage is most pronounced in Temporal tasks. Here, our method surpasses the best baseline score of 44.12 to reach 48.97. Furthermore, our approach exceeds the strong *Session-based* baseline of 36.59. This demonstrates that dynamic state switching effectively captures fine-grained semantic shifts within a single session, whereas coarse-grained session storage fails to distinguish these nuances. Finally, to ensure practical robustness, we evaluated the system under varying boundary detection error rates. As detailed in Appendix D.3, our framework maintains high stability. It consistently outperforms fixed-window baselines even with 40% segmentation noise.

4 Conclusion

In this paper, we proposed GAM, a hierarchical Graph-based Agentic Memory framework that structurally mitigates the conflict between rapid context perception and stable knowledge retention. By decoupling narrative buffering from semantic consolidation through a state-based mechanism triggered by semantic divergence, our approach ensures that memory updates occur only at semantically complete boundaries, isolating transient noise. This architecture, combined with a graph-guided retrieval strategy fusing temporal, confidence, and role-centric signals, demonstrates superior reasoning accuracy and efficiency compared to state-of-the-art baselines. We hope this work inspires research to move beyond static retrieval-optimized systems toward autonomous, self-organizing memory paradigms capable of supporting lifelong learning in complex, open-ended environments.

REFERENCES

Petr Anokhin, Nikita Semenov, Artyom Sorokin, Dmitry Evseev, Andrey Kravchenko, Mikhail Burtsev, and Evgeny Burnaev. Arigraph: Learning knowledge graph world models with episodic memory for llm agents. *arXiv preprint arXiv:2407.04363*, 2024.

Abrar Anwar, John Welsh, Joydeep Biswas, Soha Pouya, and Yan Chang. Remembr: Building and reasoning over long-horizon spatio-temporal memory for robot navigation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2838–2845. IEEE, 2025.

Hongyu Chang, Wenbo Tang, Annabella M Wulf, Thokozile Nyasulu, Madison E Wolf, Antonio Fernandez-Ruiz, and Azahara Oliva. Sleep microstructure organizes memory replay. *Nature*, 637 (8048):1161–1169, 2025.

Ding Chen, Simin Niu, Kehang Li, Peng Liu, Xiangping Zheng, Bo Tang, Xinchi Li, Feiyu Xiong, and Zhiyu Li. Halumem: Evaluating hallucinations in memory systems of agents. *arXiv preprint arXiv:2511.03506*, 2025.

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.

Aniket Didolkar, Anirudh Goyal, Nan Rosemary Ke, Siyuan Guo, Michal Valko, Timothy Lillicrap, Danilo Jimenez Rezende, Yoshua Bengio, Michael C Mozer, and Sanjeev Arora. Metacognitive capabilities of llms: An exploration in mathematical problem solving. *Advances in Neural Information Processing Systems*, 37:19783–19812, 2024.

Yufeng Du, Minyang Tian, Srikanth Ronanki, Subendhu Rongali, Sravan Bodapati, Aram Galstyan, Azton Wells, Roy Schwartz, Eliu A Huerta, and Hao Peng. Context length alone hurts llm performance despite perfect retrieval. *arXiv preprint arXiv:2510.05381*, 2025.

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

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*, 2024.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, et al. Memory in the age of ai agents. *arXiv preprint arXiv:2512.13564*, 2025.

Zixi Jia, Qinghua Liu, Hexiao Li, Yuyan Chen, and Jiqiang Liu. Evaluating the long-term memory of large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19759–19777, 2025.

Jinhao Jiang et al. Structrag: Boosting knowledge intensive reasoning of llms via hybrid information structuring. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

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

Jiho Kim, Woosog Chay, Hyeonji Hwang, Daeun Kyung, Hyunseung Chung, Eunbyeol Cho, Yohan Jo, and Edward Choi. Dialsim: A real-time simulator for evaluating long-term multi-party dialogue understanding of conversation systems. *arXiv preprint arXiv:2406.13144*, 2024.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. A human-inspired reading agent with gist memory of very long contexts. *arXiv preprint arXiv:2402.09727*, 2024.

Chuangtao Ma, Yongrui Chen, Tianxing Wu, Arijit Khan, and Haofen Wang. Large language models meet knowledge graphs for question answering: Synthesis and opportunities. *arXiv preprint arXiv:2505.20099*, 2025.

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.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems, 2024. 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*, pp. 1–22, 2023.

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.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, et al. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8416–8439, 2025.

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*, 2025.

Enneng Yang, Anke Tang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, and Jie Zhang. Continual model merging without data: Dual projections for balancing stability and plasticity. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Shiyi Yang, Zhibo Hu, Xinshu Li, Chen Wang, Tong Yu, Xiwei Xu, Liming Zhu, and Lina Yao. Drunkagent: Stealthy memory corruption in llm-powered recommender agents. *arXiv preprint arXiv:2503.23804*, 2025b.

Guibin Zhang, Muxin Fu, Guancheng Wan, Miao Yu, Kun Wang, and Shuicheng Yan. G-memory: Tracing hierarchical memory for multi-agent systems. *arXiv preprint arXiv:2506.07398*, 2025.

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.

A Related Work

A.1 Unified Stream-based Memory

Current LLM-based agents primarily adopt a unified stream paradigm to maximize interaction plasticity (Hu et al., 2025). Early architectures like Generative Agents (Park et al., 2023) record observations in a linear stream, relying on retrieval to reconstruct context. To manage infinite context windows, systems such as MemGPT (Packer et al., 2024), MemoryOS (Kang et al., 2025), and Mem0 (Chhikara et al., 2025) introduce hierarchical tiers inspired by operating systems, swapping information between working context and external storage. While these methods excel at rapid information encoding, they fundamentally lack write isolation. Newly acquired and often noisy information is directly appended or merged into the long-term store. This open-gate policy makes them susceptible to Memory Contamination (Chen et al., 2025), where continuous updates cause semantic drift (Yang et al., 2025a) or memory loss. Even advanced self-reflective agents like A-Mem (Xu et al., 2025) trigger updates based on arbitrary token counts or time-steps rather than semantic completeness, failing to prevent the corruption of stable knowledge by transient dialogue states.

A.2 Discrete Structured Memory Architectures

In contrast, Discrete Structured Memory Architectures prioritize stability through rigid knowledge representations. Systems like GraphRAG (Edge et al., 2024) and StructRAG (Jiang et al., 2025) construct static knowledge graphs to facilitate precise multi-hop reasoning. Approaches such as LightRAG (Guo et al., 2024) and G-Memory (Zhang et al., 2025) further enhance this by indexing dual-level graph structures for complex entity retrieval. However, these architectures often suffer from rigidity and latency. The expensive index construction process required by traditional knowledge graphs makes them poorly suited for the fluid narrative flow of open-domain dialogue (Ma et al., 2025). Consequently, they often produce fragmented discourse representations that excel at static fact retrieval but struggle to capture the continuous evolution of user states or sub-topic nuances in real-time.

A.3 Approaches for Structural Dynamism

Recent research has attempted to bridge the gap between stability and plasticity by introducing dynamic elements to structured memory. AriGraph (Anokhin et al., 2024) introduces episodic nodes to track changing states but is tailored for text-based games with discrete state transitions, which are rarely present in the ambiguous boundaries of natural conversation. Its world-modeling emphasis assumes that state updates are externally identifiable, whereas GAM centers on the decision of when noisy interaction fragments should be consolidated into a stable structure. Similarly, Zep (Rasmussen et al., 2025) addresses dynamism by focusing on optimizing the retrieval infrastructure rather than resolving the cognitive conflict between encoding and consolidation. Unlike these approaches, which rely on artificial state definitions or purely retrieval-based optimizations, our framework explicitly decouples the memory lifecycle, ensuring that global memory is updated only through consolidated and semantically complete units.

B Implementation Details

B.1 End-to-End Execution Flow

For completeness, we summarize the online maintenance and inference workflow of GAM below.

B.2 Event-Driven Boundary Detection

To improve reproducibility, we summarize the operational trigger logic used by the semantic boundary detector. Rather than scanning every incoming turn, GAM maintains a bounded episodic buffer and invokes the discriminator only when a sparse maintenance event occurs.

Algorithm 1 The GAM Execution Flow: Memory Maintenance and Retrieval

Require: Dialogue stream $D = \{u_1, \dots\}$, Query q, Thresholds ϵ, τ
Ensure: Response to q, Updated Memory $\mathcal{H}_t$
Process I: Memory Maintenance (Online)
for each incoming utterance u_t in D **do**
  Update the active event graph with e_t by Eq. 6
  Compute boundary indicator b_t by Eq. 2
  if $b_t = 1$ **then**
    Generate topic node v_{new} by Eq. 7
    Update the topic graph by Eq. 8
    Archive buffer: $\mathcal{S}_{\text{arch}}^{(t)} \leftarrow \mathcal{S}_{\text{arch}}^{(t-1)} \cup \{\mathcal{G}_{\text{event}}^{(t)}\}$
    Establish cross-layer index: $\mathcal{E}_{\text{cross}}^{(t)} \leftarrow \mathcal{E}_{\text{cross}}^{(t-1)} \cup \{(v_{\text{new}}, \mathcal{G}_{\text{event}}^{(t)})\}$
    Reset buffer: $\mathcal{G}_{\text{event}}^{(t)} \leftarrow \emptyset$
  end if
end for
Process II: Graph-Guided Retrieval (Inference)
Identify top-k anchors $\mathcal{V}_{\text{top}}$ from $\mathcal{G}_{\text{topic}}$ via similarity
Expand scope to semantic anchors $\mathcal{V}_{\text{anchor}}$ by Eq. 9
Extract candidate set $\mathcal{C}$ by Eq. 10
Rank candidates $v \in \mathcal{C}$ by Eq. 11
return Top-ranked context for generation

Algorithm 2 Event-Driven Semantic Boundary Detection

Require: Incoming turn u_t, token limit $T_{\max} = 2048$
Append u_t to the local buffer B_t and update the active event graph as in Eq. 6
if SessionEnd(u_t) or TokenCount(B_t) $> T_{\max}$ **then**
  Estimate the proxy boundary indicator b_t for Eq. 2 via $\text{LLM}_{\text{detect}}(\text{Prompt}_{\text{boundary}}, B_t)$
  if $b_t = 1$ **then**
    Trigger semantic consolidation for the buffered content
  end if
  Reset or trim the local buffer according to the consolidation result
end if

B.3 Coarse-to-Fine Topic Linking

The consolidation stage does not score a new semantic node against every existing topic node. Instead, GAM narrows the candidate set through vector retrieval and then applies LLM-based relation scoring only to a small candidate pool.

Algorithm 3 Coarse-to-Fine Candidate Selection for Topic Linking

Require: New semantic node v_{new}, global graph $\mathcal{G}_{\text{topic}}$, retrieval depth $K = 5$, threshold τ
Retrieve top-K candidates using the summary field c_{sum} from Eq. 7
$C \leftarrow \text{VectorRetrieve}(\mathcal{G}_{\text{topic}}, c_{\text{sum}}, \text{top\_k} = K)$
for each topic node v in C **do**
  $(w, r) \leftarrow \text{LLM}_{\text{score}}(\text{Prompt}_{\text{relation}}, v_{\text{new}}, v)$
  if $w > \tau$ **then**
    Add edge (v_{new}, v) with relation r and weight w
  end if
end for
Update the topic graph by Eq. 8

C Detailed Experimental Results

C.1 Detailed Analysis of LongDialQA Results

Table 5 presents the comprehensive performance breakdown on the LongDialQA benchmark across three distinct sub-datasets: *The Big Bang Theory*, *Friends*, and *The Office*. These datasets represent highly complex multi-party environments with frequent speaker switching and interleaved plotlines.

Table 5: Detailed performance comparison on the LongDialQA dataset, including the newly added Mem0 baseline. We report F1 and BLEU-1 scores on three sub-datasets (Big Bang Theory, Friends, The Office) and their average. The best results for each backbone are highlighted in bold, and the second best are underlined.

Model	Method	Big Bang Theory		Friends		The Office		Avg	
		F1	B-1	F1	B-1	F1	B-1	F1	B-1
Llama 3.2-3B	ReadAgent	5.19	4.70	3.49	4.24	3.24	3.18	3.97	4.04
	Memorybank	4.77	4.46	4.14	3.77	2.87	2.56	3.93	3.60
	Memgpt	3.94	2.97	3.99	3.85	3.77	3.25	3.90	3.36
	A-Mem	5.69	4.20	3.79	3.32	2.86	2.29	4.11	3.27
	MemoryOS	8.65	6.46	2.98	3.56	5.92	3.35	5.85	4.46
	Mem0	**10.84**	6.63	3.50	5.37	6.20	5.25	6.85	5.75
	Ours	9.42	**8.85**	**4.83**	**5.70**	**7.82**	**5.99**	**7.36**	**6.85**
Qwen 2.5-7B	ReadAgent	11.67	11.40	2.74	3.18	4.83	4.46	6.41	6.35
	Memorybank	9.11	9.15	7.29	7.81	5.27	4.18	7.22	7.05
	Memgpt	6.22	5.87	5.23	4.36	3.37	3.05	4.94	4.43
	A-Mem	7.93	8.09	4.85	5.87	3.68	3.42	5.49	5.79
	MemoryOS	9.71	8.48	5.71	3.20	4.86	3.64	6.76	5.11
	Mem0	15.01	14.97	6.69	6.14	9.11	8.63	10.27	9.91
	Ours	**18.73**	**18.35**	**7.32**	**8.22**	**11.60**	**10.72**	**12.55**	**12.43**
Qwen 2.5-14B	ReadAgent	13.85	13.79	7.11	7.71	5.18	5.05	8.71	8.85
	Memorybank	17.58	17.56	4.60	6.03	7.41	7.12	9.86	10.24
	Memgpt	8.99	8.87	3.91	4.14	4.60	4.43	5.83	5.81
	A-Mem	14.88	14.04	6.48	5.78	4.23	3.98	8.53	7.93
	MemoryOS	16.36	15.70	6.38	5.91	4.48	3.56	9.07	8.39
	Mem0	**18.32**	17.58	7.22	7.68	8.90	7.57	11.48	10.94
	Ours	18.13	**17.73**	**7.36**	**8.03**	**10.10**	**9.22**	**11.86**	**11.66**
GPT-4o-mini	ReadAgent	6.65	5.89	3.12	1.18	3.73	2.40	4.50	3.16
	Memorybank	7.98	8.21	6.41	7.42	3.46	2.54	5.95	6.06
	Memgpt	5.27	4.80	1.83	2.29	4.52	2.75	3.87	3.28
	A-Mem	6.22	5.60	6.80	7.29	6.44	5.01	6.49	5.97
	MemoryOS	13.36	11.63	9.56	8.16	8.92	8.13	10.61	9.31
	Mem0	13.73	12.77	9.42	8.28	**9.54**	**8.46**	10.90	9.84
	Ours	**14.70**	**13.70**	**9.64**	**8.34**	9.21	8.39	**11.18**	**10.14**

Our framework demonstrates consistent superiority over state-of-the-art baselines across most configurations. Specifically, on the *Friends* and *The Office* sub-datasets, which are characterized by nuanced interpersonal dynamics and long-term character arcs, our method achieves significantly higher F1 scores compared to the strong commercial baseline Mem0. For instance, using the Qwen 2.5-7B backbone, our approach surpasses Mem0 by a notable margin on the *The Office* subset (11.60 vs 9.11). This indicates that the proposed Role-Centric Contextualization factor effectively disentangles narrative threads in dense multi-speaker scenarios, preventing the confusion often observed in OS-based or heuristic memory systems.

Furthermore, the results highlight the model-agnostic robustness of our architecture. While baselines like A-MEM and MemoryOS exhibit performance fluctuations when scaling down from GPT-4o-mini to Llama 3.2-3B, our framework maintains a stable performance advantage. Even on the smaller Llama 3.2-3B model, our method achieves the highest Average F1 score (7.36), suggesting that the structured graph memory provides a crucial external knowledge scaffold that compensates for the limited internal context capacity of smaller LLMs.

C.2 Detailed Ablation Study Results

Table 6 presents the comprehensive breakdown of our ablation study on the LoCoMo dataset. We evaluate the contribution of each component across four specific reasoning categories: Multi-Hop, Temporal, Open-Domain, and Single-Hop.

Table 6: Detailed ablation study results across different task categories on the LoCoMo dataset.

Method	Multi-Hop		Temporal		Open-Domain		Single-Hop		Avg	
	F1	B-1	F1	B-1	F1	B-1	F1	B-1	F1	B-1
w/o TAN	31.80	21.29	39.74	34.15	16.28	13.42	52.46	47.12	35.07	29.00
w/o MFR	33.73	22.91	46.12	38.06	16.30	14.20	47.61	41.96	35.94	29.28
w/o SSM	30.32	19.96	39.53	32.55	16.73	14.08	43.72	37.93	32.58	26.13
w/o EPG	25.35	18.94	33.61	27.70	14.59	13.44	26.68	22.97	25.06	20.76
Ours	**35.32**	**25.77**	**48.97**	**39.68**	**21.14**	**17.89**	**54.58**	**48.62**	**40.00**	**32.99**

C.3 Comparison with AriGraph on LoCoMo

To further position GAM against graph-based memory approaches with explicit world modeling, we implement AriGraph as an additional baseline on LoCoMo. While AriGraph is effective when state transitions are externally identifiable, open-domain dialogue presents fuzzy topic boundaries and temporally entangled evidence, making semantic consolidation substantially more challenging.

Table 7: Performance comparison with AriGraph on the LoCoMo benchmark. We report F1 and BLEU-1 for each reasoning category and their average.

Model	Method	Multi-Hop		Temporal		Open-Domain		Single-Hop		Avg	
		F1	B-1	F1	B-1	F1	B-1	F1	B-1	F1	B-1
Llama 3.2-3B	AriGraph	13.99	7.83	3.72	2.69	20.10	17.74	14.63	11.67	12.58	9.47
	Ours	**28.74**	**22.35**	**46.12**	**37.24**	**21.18**	**18.33**	**49.02**	**42.18**	**36.27**	**30.02**
Qwen 2.5-7B	AriGraph	21.32	12.30	5.51	5.25	15.10	12.40	24.44	19.49	19.34	14.77
	Ours	**35.32**	**25.77**	**48.97**	**39.68**	**21.14**	**17.89**	**54.58**	**48.62**	**40.00**	**32.99**
Qwen 2.5-14B	AriGraph	29.40	18.20	6.00	5.01	16.87	13.65	30.68	24.38	24.44	18.54
	Ours	**33.32**	**25.38**	**50.25**	**39.33**	**20.40**	**18.50**	**57.55**	**50.99**	**40.38**	**33.55**
GPT-4o-mini	AriGraph	25.10	12.92	8.44	6.00	15.32	11.53	30.93	24.52	24.20	17.72
	Ours	**35.88**	**27.96**	**51.96**	**42.26**	**28.12**	**24.51**	**56.58**	**51.18**	**43.14**	**36.48**

Across all backbones, GAM consistently surpasses AriGraph, with the largest gap appearing in Temporal reasoning. This supports the claim that a semantic-boundary-triggered memory lifecycle is better suited to continuous dialogue streams than graph updates designed for explicitly observable discrete states.

C.4 MovieChat-1K Proxy Multimodal Experiment

To probe the modality-agnostic structure of GAM without redesigning the full system into a native multimodal model, we conducted a proxy experiment on MovieChat-1K using dense video captions as a continuous visual narrative stream. We compare GAM against a long-context direct-input baseline under the Qwen2.5-7B backbone with retrieval depth $K = 15$. The same temporal, role, and confidence retrieval factors used for text were transferred in a zero-shot manner by aligning them with timestamps, on-screen characters, and action coherence.

The proxy results suggest that the hierarchical graph memory and state-based consolidation mechanism are not tied to textual dialogue alone. Even without domain-specific tuning, GAM improves both accuracy and F1 on long-form visual narratives represented as caption streams, supporting the multimodal roadmap discussed in the limitations section.

Table 8: Proxy multimodal evaluation on MovieChat-1K using dense video captions as the input stream.

Method	Accuracy	F1 Score	BLEU-1
Baseline (Direct Input)	47.35	42.35	45.31
Ours (GAM)	**55.51**	**48.95**	**51.63**
Improvement	+8.16	+6.60	+6.32

D System Analysis

D.1 Operational Cost of Boundary Detection

The efficiency table in the main text reports query-time inference cost. To complement it, we separately measure the maintenance-stage cost of the semantic boundary detector on LoCoMo. Because the detector is triggered sparsely rather than at every turn, the additional overhead remains small relative to the main inference pipeline.

Table 9: Cost analysis of the semantic divergence detection mechanism on LoCoMo.

Metric	Value
Avg. Sessions per Sample	27.2
Avg. Latency per Session	0.63 s
Avg. Input Tokens per Session	932.36
Avg. Output Tokens per Session	6.15

These results confirm that the detector behaves as a sparse maintenance primitive rather than a high-frequency online bottleneck. Combined with the event-driven trigger logic in Appendix B, this explains why GAM can preserve efficiency despite relying on LLM-based semantic boundary discrimination.

D.2 Long-Term Scaling Complexity

In addition to the retrieval-size sensitivity analysis, we analyze how the graph grows under progressive session injection. The key question is whether increasing long-term memory size causes inference to scale poorly. We therefore incrementally inject up to 27 long-horizon sessions and track structural growth together with inference cost.

Table 10: Complexity evolution of the dual-layer graph memory under progressive session injection.

Sessions	Event Nodes	Topic Nodes	Edges	Avg. Tokens	Avg. Latency (s)	Avg. F1
3	81	8	371	1604	3.49	0.65
15	363	43	1791	1864	4.16	0.57
27	657	84	3307	1836	4.71	0.48

Although the total number of graph edges increases substantially, the inference token count and latency remain comparatively stable because retrieval first prunes the search space through the topic layer before drilling down into episodic evidence. This supports the claim that GAM avoids uncontrolled growth in query-time overhead even as the long-term memory structure expands.

D.3 Robustness Analysis on Topic Segmentation Noise

D.3.1 Experimental Setup

To rigorously evaluate the robustness of our GAM framework against errors in the topic boundary detection module $\mathcal{M}_\theta$, we designed a composite noise injection mechanism that simulates three distinct types of segmentation failures. These include Miss errors which correspond to false negatives, Shift errors resulting in temporal misalignment, and Extra errors serving as false positives.

Specifically, given a noise level η ranging from 0.0 to 0.4, we first iterate through the ground-truth boundaries. With a probability of η, each boundary is either deleted to simulate a missed topic switch or shifted randomly by up to two turns in either direction to simulate imprecise boundary localization.

Subsequently, to simulate hallucinated boundaries, we inject extra random cut-points into the dialogue where the number of inserted boundaries is proportional to the original count scaled by η. This setup creates a challenging and highly perturbed environment that tests the system's ability to recover context despite fragmentation, fusion, and misalignment errors.

D.3.2 Results and Analysis

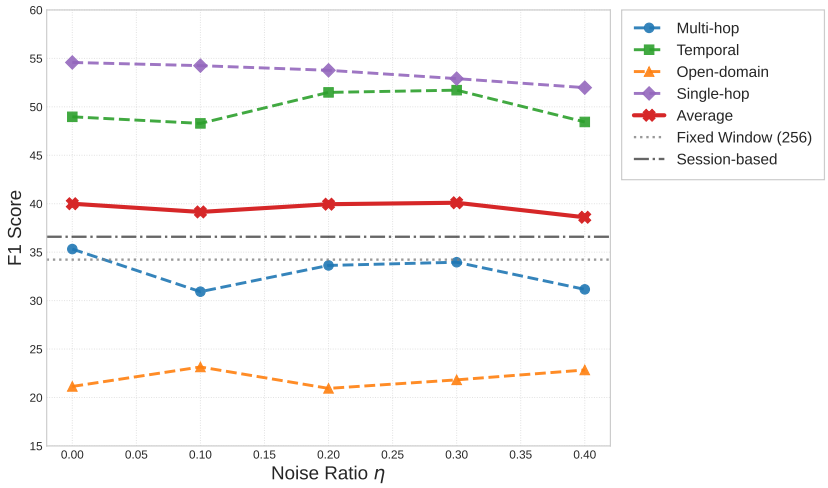

Figure 4: Robustness analysis of the proposed framework under varying topic segmentation noise levels η. The solid red line represents the Average F1 score, which remains remarkably stable around 40.0, significantly outperforming the Fixed Window (Gray Dotted Line) and Session-based (Black Dash-Dotted Line) baselines even at $\eta = 0.4$. Dashed lines indicate performance on specific task categories, highlighting the resilience of the hierarchical graph architecture against boundary errors.

As shown in Figure 4, the performance trajectory on the LoCoMo benchmark demonstrates the exceptional stability of our framework against segmentation errors. Despite complex perturbations including boundary deletions and shifts, the Average F1 score remains consistent around 40.0. Even at the most severe noise level where η equals 0.4, the model achieves an F1 score of 38.60, which significantly outperforms both the Fixed Window baseline score of 34.23 and the Session-based baseline score of 36.59. These results confirm that the efficacy of our method is not contingent on perfect segmentation, as the underlying Hierarchical Graph Architecture provides a robust scaffolding that organizes information effectively even when structural boundaries are imprecise.

A granular analysis reveals divergent impacts across task categories. Intriguingly, Temporal tasks showed improved performance under moderate noise conditions, peaking at 51.72 when η is set to 0.3. This suggests that the random insertions of extra boundaries create an over-segmentation effect, producing finer-grained memory nodes that facilitate precise temporal grounding. In contrast, Single-hop retrieval exhibited a graceful degradation from 54.58 to 51.98, as the shifting or deletion of boundaries can disrupt the local coherence required for keyword matching. However, this decline remains minimal at less than 5%, demonstrating that our Graph-Guided Multi-Factor Retrieval strategy effectively compensates for structural misalignments by utilizing the semantic connectivity of the graph to bridge gaps created by segmentation errors.

E HYPERPARAMETER ANALYSIS

E.1 MODULATION FACTORS SENSITIVITY ANALYSIS

To evaluate the robustness of our Graph-Guided Multi-Factor Retrieval mechanism, we conducted a sensitivity analysis on the modulation factors β_{role}, β_{time}, and β_{conf}. We varied each factor from 1.0 to 2.0 while keeping others constant. The results are visualized in Figure 5.

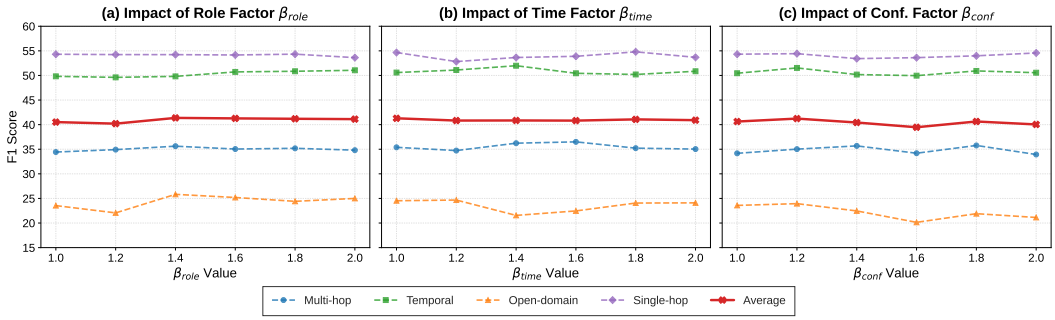

Figure 5: Parameter sensitivity analysis on the LoCoMo dataset using the Qwen 2.5-7B backbone. The three subplots illustrate the impact of varying the (a) Role Factor β_{role}, (b) Temporal Factor β_{time}, and (c) Confidence Factor β_{conf} on retrieval performance across different task categories. The solid red line represents the Average F1 Score. The relatively flat curves indicate the system's robustness to hyperparameter variations.

Impact of Role Factor (β_{role}). As illustrated in Figure 5(a), the performance generally improves as the role factor increases, peaking at approximately $\beta_{role} = 1.4$. This trend is particularly evident in the Multi-hop task (blue dashed line), confirming that filtering memory candidates based on speaker identity helps the model connect disparate pieces of information associated with specific characters. However, setting the value too high ($\beta > 1.6$) leads to a slight decline, likely because excessive filtering might exclude relevant context mentioned by other speakers in the same scene.

Impact of Temporal Factor (β_{time}). Figure 5(b) demonstrates a distinct trade-off. The Temporal reasoning task (green dashed line) benefits significantly from a higher boosting factor, reaching optimal performance around $\beta_{time} = 1.4$. Conversely, the Open-domain task (orange dashed line) experiences a dip at this value. This suggests that while strong temporal modulation sharpens the focus for time-sensitive queries, it may occasionally narrow the retrieval scope for broader, non-temporal questions. We selected 1.4 as the default setting to maximize temporal precision while maintaining a high Average F1 score.

Impact of Confidence Factor (β_{conf}). As shown in Figure 5(c), the system exhibits high stability with respect to the intrinsic confidence factor. The curves remain relatively flat across the tested range, with a slight peak observed between 1.2 and 1.4. This indicates that while prioritizing high-quality memories (those passing the self-test) contributes to overall reliability, the underlying topological structure of the graph plays a more dominant role in retrieval success than the precise tuning of this weight.

Overall, the sensitivity analysis confirms that our multi-factor retrieval mechanism is robust. The standard deviation of the Average F1 score across all tested parameter ranges remains low, implying that the framework's effectiveness derives primarily from its architectural design rather than over-optimized hyperparameters.

E.2 RETRIEVAL SIZE SENSITIVITY ANALYSIS

The retrieval size k serves as a critical hyperparameter determining the volume of recalled context. To investigate its impact, we conducted a sensitivity analysis on the LoCoMo dataset, varying k from 5 to 40.

As shown in Figure 6, the results demonstrate that performance rapidly improves, reaching a peak at $k = 10$, but subsequently stagnates or degrades as k increases to 40. This decline suggests that an overly large retrieval set introduces irrelevant noise, which disrupts the model's reasoning. Notably, *Temporal* tasks are particularly sensitive to this interference, requiring focused windows for chronological precision, whereas *Multi-hop* tasks remain robust across larger contexts. To balance performance and efficiency, we select $k = 10$ as the optimal setting, achieving peak accuracy with minimal token consumption compared to larger k.

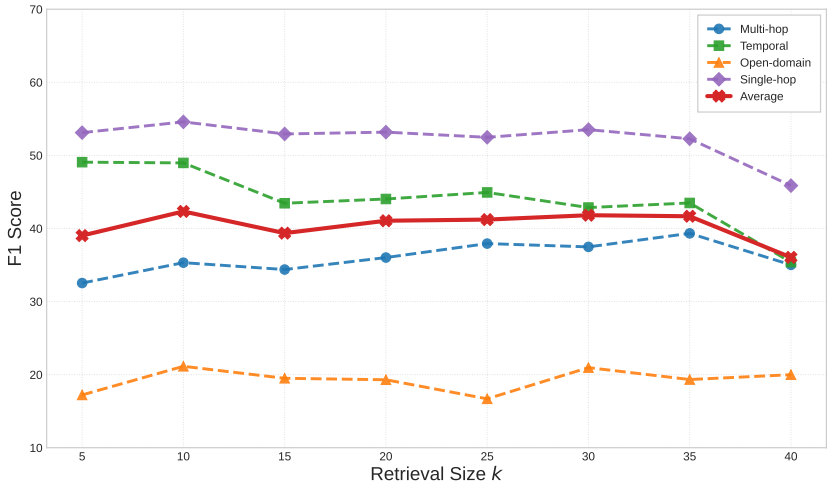

Figure 6: Impact of retrieval size k on model performance across different task categories. We select $k = 10$ as the optimal setting to balance reasoning accuracy with computational efficiency.

F Prompt Engineering Details

To ensure the reproducibility of our experiments, we provide the exact instruction prompts used in the key modules of our framework. All prompts are designed to output structured JSON data to facilitate downstream programmatic processing.

F.1 Semantic Relation & Edge Weighting Prompt

Semantic Relation & Edge Weighting Prompt

```
Determine the relationship between the following two memories.
Memory 1: {memory_1_content}
Memory 2: {memory_2_content}
Possible relations:
- "support": one memory supports the other (asymmetric)
- "contradict": one memory contradicts the other (asymmetric)
- "coreference": same entity or event (symmetric)
- "causal": one memory leads to the other (asymmetric)
- "semantic": similar meaning (symmetric)
- "unrelated": no meaningful relation
Return JSON strictly in the format: {{"relation": "...", "confidence": 0-1}}
```

To construct the global Topic Associative Network in §2.3, we use this prompt to judge the edge type and weight between nodes. The confidence score (0-1) output by the LLM is directly utilized as the edge weight, allowing the retrieval mechanism to prioritize high-confidence associations.

F.2 Semantic Node Summarization Prompt

During the Semantic Consolidation State (§2.3), we generate the dual-granularity representation for new topic nodes. This prompt generates the c_{sum} attribute. We specifically ask for "keywords" and a "concise summary" separately to support both lexical matching and semantic embedding retrieval.

Semantic Node Summarization Prompt

```
Generate a structured analysis of the following content by:
1. Identifying the most salient keywords and core themes (focus on nouns, verbs, and
   key concepts).
2. Writing a concise summary (one to two sentences). Don't be redundant, summarize
   everything in the fewest words possible.
Format the response as a JSON object:
{
   "keywords": [
     // several specific, distinct keywords
     // Order from most to least important
   ],
   "summary": a concise one to two sentence summary
}
Content for analysis:{content}
```

F.3 Topic Boundary Detection Prompt

This prompt corresponds to the discriminator function f_θ described in §2.3. We employ a sliding window strategy to provide context. The prompt explicitly instructs the model to return indices in a JSON array to ensure robust parsing by the state machine.

Topic Boundary Detection Prompt

```
Please analyze the following conversation and identify where topic changes occur. A
topic change happens when the conversation shifts to a different subject or theme.
Conversation:{dialogue_text}
Identify the indices (positions) where topic boundaries occur. A boundary at index i
means that the topic changes between utterance i and utterance i+1. Return only the
boundary indices as a JSON array.
Return your response as a JSON object with exactly this structure:
{{ "boundaries": [array_of_indices] }}
Example: {{ "boundaries": [2, 6] }}
```

G Limitations

Although our experiments on LoCoMo and LongDialQA verify the effectiveness of the proposed architecture, the current instantiation of GAM is limited to text modality. This unimodal design limits the system's ability to leverage the rich nonverbal information commonly found in real-world interactions, such as visual context or acoustic cues. Consequently, the model cannot perform fine-grained reasoning over visual or auditory inputs. A native multimodal extension would replace purely textual memory representations with multimodal nodes that combine symbolic summaries with modality-specific features such as keyframe or audio representations. The same state-switching mechanism could then detect semantic shifts from visual or acoustic signals instead of only lexical topic changes. We leave this extension to future work and do not claim native multimodal reasoning in the current system.

H BROADER IMPACTS AND SAFETY CONSIDERATIONS

Persistent memory systems for conversational agents raise meaningful privacy and safety concerns. First, users may reveal sensitive personal details that should not automatically enter long-term storage. GAM does not solve this issue by itself, but its state-based consolidation design creates a natural intervention point: before buffered content is written into the global topic graph, privacy filters or user-defined retention rules can inspect and block sensitive content.

Second, long-term memory systems should support user control over what is stored. Because GAM organizes memory as explicit graph nodes and cross-layer links rather than only opaque vector states, it offers a clearer substrate for inspection, selective deletion, freezing, or branch-level pruning. This does not guarantee that a deployed system will expose such controls, but it makes them architecturally easier to implement.

Third, memory-bearing agents may form inaccurate beliefs about users through faulty summarization or incorrect semantic links. In a graph-based memory, such failures can in principle be corrected locally by removing spurious edges, adjusting confidence weights, or deleting incorrect topic nodes, without retraining the entire backbone model. We view these affordances as useful safety primitives, not as complete solutions; robust deployment still requires careful interface design, auditing, and user consent mechanisms.