MemORAI: Memory Organization and Retrieval via Adaptive Graph Intelligence for LLM Conversational Agents

Hung Pham Van[1]*, Nguyen Manh Hieu[1]*, Khang Pham Tran Tuan[2]*, Nam Le Hai[2,†], Linh Ngo Van[2], Diep Thi-Ngoc Nguyen[3], Trung Le[4]

[1]Independent Researcher, [2]Hanoi University of Science and Technology, [3]VNU University of Engineering and Technology, [4]Monash University

Abstract

Large Language Models (LLMs) lack persistent memory for long-term personalized conversations. Existing graph-based memory systems suffer from information dilution, absent provenance tracking, and uniform retrieval that ignores query context. We introduce MemORAI (Memory Organization and Retrieval via Adaptive Graph Intelligence), a framework that integrates three innovations: selective memory filtering with dual-layer compression to retain user-persona-relevant content, a provenance-enriched multi-relational graph tracking factual origins at the turn level, and query-adaptive subgraph retrieval with Dynamic Weighted PageRank that applies query-conditioned edge weighting. Evaluated on `LOCOMO` and `LongMemEval` benchmarks, MemORAI achieves state-of-the-art performance in memory retrieval and personalized response generation, demonstrating that selective storage, enriched representation, and adaptive retrieval are essential for coherent, personalized LLM agents.

1 Introduction

Human cognition relies on a dynamic memory system that balances acquisition, consolidation, and retrieval to sustain personalized interactions without cognitive overload (Liu et al., 2025). Large Language Models (LLMs), despite excelling in reasoning and generation (Team et al., 2025; Grattafiori et al., 2024; Yang et al., 2025; DeepSeek-AI et al., 2025), lack this persistence. Constrained by limited context windows, they lose crucial details (Liu et al., 2023) and reset to a stateless baseline across sessions (Timoneda and Vera, 2025; Yuan et al., 2024), making ephemeral prompting a fragile substitute that amplifies hallucinations (Lewis et al., 2021).

Memory-augmented approaches address this through external stores and selective retrieval (Liu et al., 2025; Wang et al., 2024). While retrieval-augmented generation (RAG) (Lewis et al., 2021) and vector-based systems (Yuan et al., 2024; Pan et al., 2025; Tan et al., 2025) have advanced factual grounding, they struggle with relational and temporal structures (Wang et al., 2024). Graph-based representations offer richer modeling through interconnected entities and relations (Chhikara et al., 2025; Gutiérrez et al., 2025), yet existing systems reveal critical gaps: hierarchical methods like RAPTOR require expensive re-clustering (Sarthi et al., 2024), sparse graphs like Mem0g bias toward high-degree nodes (Chhikara et al., 2025), and sophisticated approaches like HippoRAG 2 propagate scores uniformly without query-conditioned adaptation (Gutiérrez et al., 2025). Crucially, no existing system filters user-persona-relevant content from generic dialogue or tracks provenance at the turn level, leading to information dilution and opacity in factual origins.

To address these limitations, we introduce **MemORAI—Memory Organization and Retrieval via Adaptive Graph Intelligence**—a framework that integrates selective memory filtering, provenance-enriched graph construction, and query-adaptive retrieval with dynamic edge weighting. Our contributions are:

- **Selective Memory Filtering:** A memory gate that retains only user-persona-relevant content while generating segment-level summaries to preserve global context, improving storage efficiency and retrieval precision.
- **Provenance-Enriched Knowledge Graph:** A heterogeneous graph architecture with entity, turn, and segment nodes featuring explicit turn-level provenance tracking for transparent auditing and fine-grained retrieval.
- **Dynamic Weighted PageRank:** A query-adaptive retrieval method that constructs focused

*Equal contribution
†Corresponding author: namlh@soict.hust.edu.vn

Findings of the Association for Computational Linguistics: ACL 2026, pages 28235–28253

subgraphs through multi-aspect search and applies query-conditioned edge weighting to prioritize semantically aligned evidence.

2 Related Work

Memory Granularity. Early retrieval-based memory systems segmented dialogue history at either the turn or session level (Yuan et al., 2024; Wang et al., 2024). While turn-level units preserve fine details, they fragment context; session-level aggregation, by contrast, introduces irrelevant noise. To balance coherence and relevance, Pan et al. (2025) proposed **SECOM**, which segments dialogue into coherent topical units and applies compression-based denoising, while Xu et al. (2025) developed **A-MEM**, constructing dynamic "atomic notes" linked by shared attributes. These works highlight that fixed granularity constrains both retrieval efficiency and adaptability, motivating structures capable of hierarchical and relational reasoning beyond flat memory units.

Structured Memory Representations. Beyond flat chunking, hierarchical and graph-based memories enable relational reasoning and associative recall. **RAPTOR** (Sarthi et al., 2024) recursively clusters and summarizes text into a multi-level tree, supporting thematic retrieval but at significant computational cost due to recursive LLM summarization and full re-clustering during updates. **Mem0g** (Chhikara et al., 2025) introduces a graph-based memory representing conversational knowledge as entity–relation triplets, facilitating multi-hop reasoning but limited by shallow semantics—nodes often store only surface names without entity descriptions, and synonym edges are defined by name similarity rather than conceptual meaning. Similarly, **HippoRAG 2** (Gutiérrez et al., 2025) employs Personalized PageRank over dense-sparse knowledge graphs for continual retrieval, yet its propagation remains uniform across edges and its synonym linking depends solely on lexical overlap between entity names. These simplifications cause brittle relational inference, synonym noise, and uniform ranking insensitive to query semantics.

Adaptive Retrieval. Recent studies explore dynamic retrieval mechanisms to enhance contextual sensitivity. **Reflective Memory Management (RMM)** (Tan et al., 2025) refines memory organization via prospective and retrospective reflection, using reinforcement feedback to adapt retrieval weights. **HippoRAG 2** extends this direction through query-conditioned Personalized PageRank, but without relation-level semantic modulation. Consequently, existing methods remain constrained by fixed propagation rules and limited personalization, often conflating high-degree node connectivity with relevance. While recent efforts in Graph RAG have begun to mitigate this structural bias by incorporating multi-aspect semantic reranking (Hieu et al., 2025), they typically apply this after traversal. MemORAI, in contrast, directly embeds query-conditioned semantic modulation into the traversal process itself.

In contrast, MemORAI addresses these limitations through three key mechanisms. First, selective memory filtering with dual-layer compression tackles information dilution by retaining only user-persona-relevant content while preserving global context through segment summaries. Second, provenance-enriched graph construction enables transparent auditing by tracking factual origins at the turn level—a capability absent in prior work. Third, query-adaptive subgraph retrieval with Dynamic Weighted PageRank overcomes uniform propagation by applying query-conditioned edge weighting, enabling context-sensitive retrieval without exhaustive graph traversal. Together, these mechanisms establish a cohesive memory lifecycle that integrates selective storage, enriched representation, and adaptive retrieval.

3 Methodology

MemORAI implements a streamlined three-phase pipeline for long-term personalized dialogue agents (Figure 1): (1) **Session Segmentation and Selective Compression**—dialogues are segmented topically, and a memory gate retains only user-relevant utterances while summarizing generic discourse for coherence (§3.1); (2) **Provenance-Enriched Graph Construction**—entity-relation triplets are extracted from retained messages and embedded in a heterogeneous graph of entities, turns, and segments with explicit turn-level provenance (§3.2); (3) **Query-Adaptive Retrieval and Generation**—multi-aspect retrieval seeds a query-focused subgraph, Dynamic Weighted PageRank ranks nodes by query-conditioned semantic alignment, and top-ranked turns with supporting triplets are formatted into provenance-aware prompts for personalized response generation (§3.3).

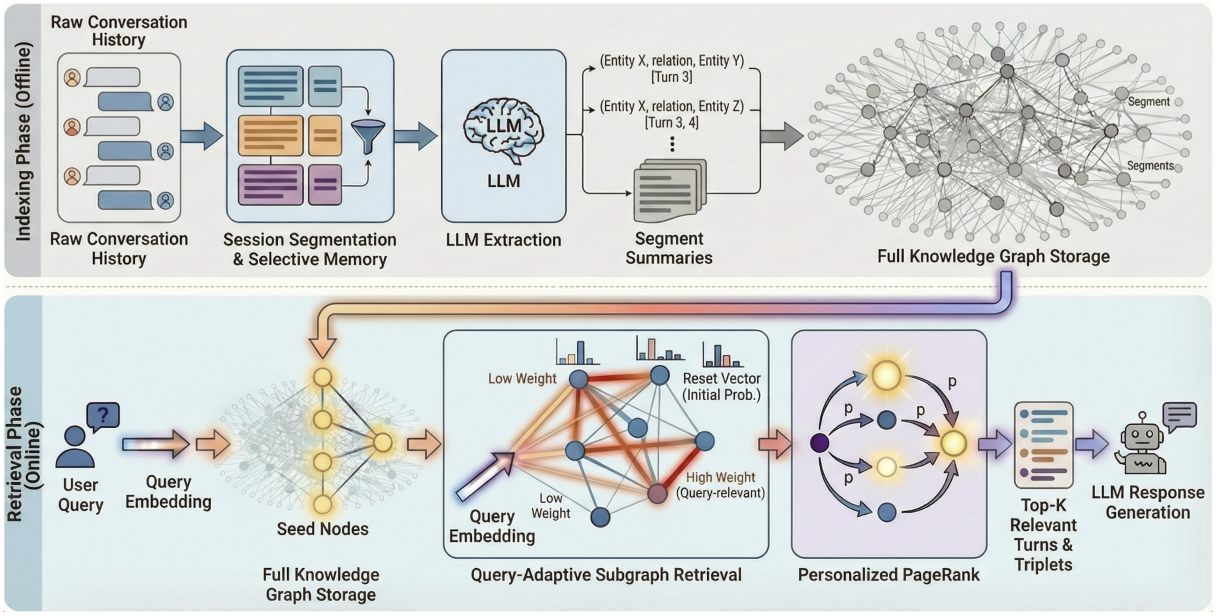

Figure 1: Overview of MemORAI's three-phase pipeline. (1) **Session Segmentation and Selective Compression**: Conversations are segmented topically; a memory gate retains user-relevant utterances and summarizes discarded content. (2) **Provenance-Enriched Graph Construction**: LLM extraction produces entity-relation triplets with turn-level provenance in a heterogeneous graph. (3) **Query-Adaptive Retrieval and Generation**: Multi-aspect search identifies seed nodes; query-focused subgraph assembly enables Dynamic Weighted PageRank with query-conditioned edge weights; top-ranked turns and triplets guide personalized response generation.

3.1 Session Segmentation & Selective Compression

Following SECOM (Pan et al., 2025), we first decompose raw multi-session conversations into semantically coherent segments $S_i = \{t_1, t_2, \ldots, t_m\}$ via LLM prompting (Appendix C.1). For each segment, we apply a selective memory gate that identifies and retains only messages containing user-specific episodic content—personal facts, preferences, commitments, and identity markers—producing a filtered set $M_i \subseteq S_i$ (Appendix C.2).

To preserve global context from discarded messages, we generate a segment-level summary σ_i via LLM prompting (Appendix C.3). This dual-layer approach stores both M_i (fine-grained personal content) and σ_i (global contextual anchor), preventing information loss while dramatically reducing storage overhead and filtering out noise. By focusing on memory-relevant content, this selective compression ensures that subsequent graph extraction and construction operate on high-quality, user-centric signals rather than generic conversational clutter.

3.2 Provenance-Enriched Graph Construction

From each filtered segment M_i, we construct a multi-relational knowledge graph $G = (V, E)$ with explicit provenance tracking. All components entities, descriptions, and triplets are extracted via LLM prompts that enforce turn-level citation (Figure 1, prompts in Appendices C.4 & C.6).

Node Types. The graph includes three node types:

- *Entity nodes* $e \in V_E$ store a `name`, fine-grained natural-language `description` (e.g., "Alex—software engineer at XYZ, prefers async communication")—an approach shown to preserve semantic details better than uniform node summaries (Hieu et al., 2025)—and `turn_ids` for provenance.
- *Turn nodes* $\tau \in V_T$ store `text`, `segment_id`, and `turn_id`.
- *Segment nodes* $s \in V_S$ store summary σ_i and `segment_id`.

  Embeddings are computed from descriptions, turn text, and summaries respectively.

Edge Types. The graph includes three edge types:

- *Entity-relation-entity edges* $e_1 \xrightarrow{r} e_2$ connect entities via typed relations (e.g., (Alex, works_at, XYZ)), storing `source_turns` for turn-level provenance.
- *Entity-turn edges* $e \leftrightarrow \tau$ link entities to their mentions.
- *Turn-segment edges* $\tau \leftrightarrow s$ preserve dialogue hierarchy.

This heterogeneous structure enables multi-hop reasoning and precise provenance-aware retrieval.

3.3 Query-Adaptive Subgraph Retrieval & Ranking

Given a user query q, MemORAI retrieves relevant memory through a two-step process: query-focused subgraph retrieval via multi-aspect seeding, followed by dynamic weighted ranking.

3.3.1 Query-Focused Subgraph Retrieval

Unlike HippoRAG 2 (Gutiérrez et al., 2025), which applies ranking across the entire memory graph, we dynamically retrieve a sparse, query-focused subgraph $G_q = (V_q, E_q)$ at query time. Through multi-aspect parallel retrieval, we first identify top-k seed nodes—both segment nodes (via summary embeddings) and entity nodes (via description embeddings) and top-k relation edges (via triplet description embeddings) using semantic similarity search. We then perform one-hop neighborhood expansion from these seeds to include all directly connected turns, entities, and segments. This query-adaptive subgraph retrieval filters out irrelevant memory regions before ranking, reducing noise that could otherwise degrade ranking performance while preserving high-quality, contextually relevant evidence with full provenance links.

3.3.2 Dynamic Weighted PageRank

Traditional PageRank algorithms prioritize nodes with many high-quality neighbors, as scores propagate recursively from authoritative sources. HippoRAG 2 (Gutiérrez et al., 2025) applies Personalized PageRank (PPR) with seed nodes extracted from queries and reset probabilities biased toward relevant starting points, enabling multi-hop reasoning through random walks over the knowledge graph. However, this uniform propagation mechanism can bias rankings toward nodes with dominant neighbor counts, potentially suppressing memory content in less-connected nodes that are nonetheless semantically relevant to the query. To address this limitation, our **Dynamic Weighted PageRank (DW-PR)** modulates score propagation based on query-conditioned edge weights that reflect semantic alignment rather than structural connectivity alone (see Figure 2). The necessity of shifting from uniform to importance-aware weighting mirrors successful strategies in recent LLM alignment and cross-tokenizer distillation, where prioritizing highly informative signals over uniform processing yields superior performance across various optimization tasks (Nguyen et al., 2026; Vu et al., 2026a; Le et al., 2025). For each edge type, we define:

$$w(u \to v) = \begin{cases} \text{sim}(q, e.\text{desc}), & u = e, v = \tau \\ \text{sim}(q, r.\text{desc}), & u \xrightarrow{r} v \\ \frac{1}{|\tau|} \sum_{e \in \tau} \text{sim}(q, e.\text{desc}), & u = \tau, v = s \end{cases} \tag{1}$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity between query and description embeddings. All nodes in the subgraph are initialized with $\text{seed}(v)$ equal to their semantic similarity to q. DW-PR scores then propagate iteratively via:

$$\text{PR}_{t+1}(v) = (1 - d) \cdot \text{seed}(v) + d \cdot S(v), \tag{2}$$

where

$$S(v) = \sum_{u \to v} \frac{w(u \to v)}{\sum_{u \to *} w(u \to *)} \text{PR}_t(u), \tag{3}$$

and d is the damping factor. This query-adaptive weighting ensures that semantically relevant but sparsely connected nodes can rank highly, preventing structural bias from overshadowing contextually critical memory content.

After convergence, turn nodes τ are ranked by their final PageRank scores, and the top-m turns are retrieved. For each retrieved turn, we also include all entity-relation triplets that cite it (i.e., where $\tau \in$ source_turns). These turns and supporting triplets are then formatted into a provenance-aware prompt that augments the conversational context, enabling the LLM to generate responses grounded in personalized memory with explicit citation of supporting evidence.

4 Experiments

4.1 Experimental Settings

Datasets & Metrics. We evaluate on two long-horizon conversational memory benchmarks: **LongMemEval-s** (Wu et al., 2025), and **LoCoMo-10** (Maharana et al., 2024). These datasets target

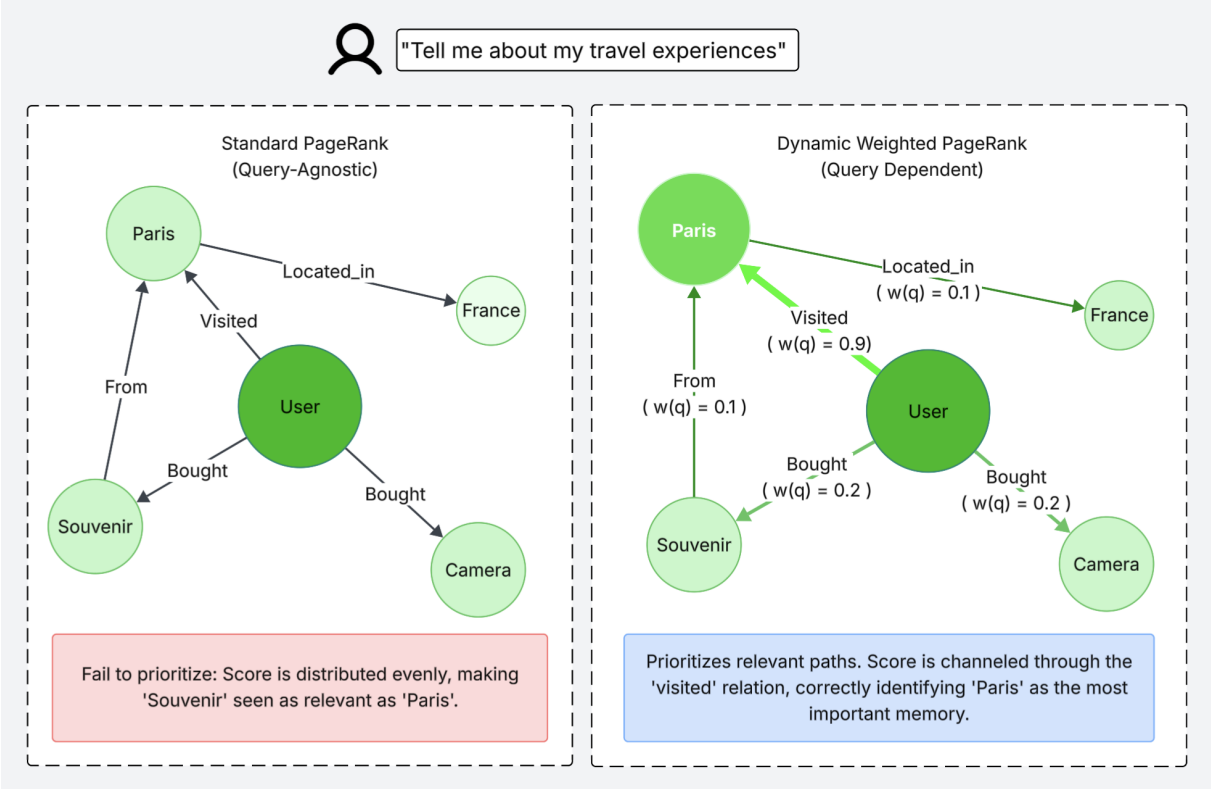

Figure 2: Traditional PageRank vs Dynamic Weighted PageRank

agent memory over sustained, multi-session dialogues rather than single-turn recall. Following an inference-only setting (no additional training or fine-tuning), we treat every QA pair in each benchmark as test data. For **retrieval evaluation**, we report Recall@k ($k \in \{3, 5, 10\}$) at both session-level and turn-level granularity. For **generation evaluation**, we measure both lexical and semantic fidelity using **F1**, **BLEU**, **ROUGE** (R1, R2, RL), and **BERTScore** (Zhang et al., 2020). We additionally employ **GPT-4o** as a judge (GPT4o-J) to assess answer correctness on a normalized scale.

Baselines. We compare our method against a diverse set of approaches spanning full-history context, dense retrieval, memory-centric conversation, and structured RAG. (1) **Full History**: uses the complete conversation records without explicit retrieval, accommodating up to a 128k-token window. **Dense retrieval models**: (2) *MPNet* (Song et al., 2020), (3) *Contriever* (Izacard et al., 2022), (4) *BGE-M3*, and (5) *BM25*. **Memory-based conversational models**: (6) *LLM-RSum* (Wang et al., 2025), which recursively summarizes and updates a compact memory buffer; (7) *MPC* (Lee et al., 2023), which leverages a pre-trained LLM to curate high-quality conversational memories; (8) *SeCom* (Pan et al., 2025), which segments long dialogues into coherent topics with compression-based denoising; (9) *MemGAS*, which combines memory gating with adaptive summarization. **Structured RAG models**: (10) *HippoRAG 2* (Gutiérrez et al., 2025), which integrates knowledge-graph indexing with graph traversal; (11) *RAPTOR* (Sarthi et al., 2024), which applies recursive summarization and hierarchical clustering; (12) *LightRAG* (Guo et al., 2024) a lightweight graph-based retrieval approach; and (13) *MemTree* (Rezazadeh et al., 2024), which organizes memories in a hierarchical tree structure.

Implementation Details. We employ `openai/gpt-oss-20b` (decoding temperature 0) uniformly across all modules and adopt top-$k = 3$ retrieval. Memory embeddings are generated with `Contriever` to ensure fair comparison with prior work—not owing to its embedding quality, but to isolate the contribution of our design and algorithmic innovations.

4.2 Main Results

We report end-to-end QA performance (Table 1), along with session- and turn-level retrieval results (Tables 2 and 3). All models—including ours—use `Contriever` for memory embeddings

Table 1: QA performance on LongMemEval-s and LOCOMO-10. GPT4o-J denotes GPT-4o judge scores (%). Best results in **bold**, second-best underlined. RAPTOR returns hierarchical summaries rather than verbatim excerpts.

LongMemEval-s							
Model	GPT4o-J	F1	BLEU	R-1	R-2	R-L	BERTScore
Full History	50.60	11.48	1.40	12.10	5.47	10.85	83.07
BM25	42.00	14.19	4.30	22.08	11.10	21.37	86.53
BGE-M3	47.60	12.30	3.82	19.19	8.80	18.56	86.14
MPNet	41.20	16.06	5.90	24.92	12.53	24.20	87.12
Contriever	41.00	23.94	9.25	32.10	15.50	30.63	88.40
LLM-RSum	35.40	12.29	2.09	13.01	5.55	11.52	83.60
MPC	53.80	13.60	1.74	14.27	6.49	12.95	83.49
SeCom	45.69	29.13	11.53	36.91	20.10	35.83	89.22
HippoRAG 2	57.60	14.73	2.15	15.30	7.36	13.83	83.86
RAPTOR	32.20	12.08	1.90	12.73	5.82	11.25	83.50
MemGAS	60.20	20.38	4.22	21.05	10.47	19.47	85.21
LightRAG	56.21	2.17	0.43	3.06	1.18	2.57	79.63
MemTree	21.80	8.57	2.27	12.02	4.38	10.44	83.49
MemOrai (Ours)	**75.55**	**45.99**	**11.54**	**50.63**	**25.25**	**50.02**	**90.37**

LOCOMO-10							
Model	GPT4o-J	F1	BLEU	R-1	R-2	R-L	BERTScore
Full History	33.43	12.23	1.84	12.70	5.66	11.73	84.07
BM25	28.05	14.19	4.30	22.08	11.10	21.36	86.53
BGE-M3	28.80	12.30	3.82	19.19	8.80	18.56	86.14
MPNet	32.93	16.05	5.90	24.92	15.29	24.19	87.12
Contriever	32.15	14.64	5.03	22.94	11.12	22.29	86.77
LLM-RSum	22.56	9.14	0.99	9.82	3.38	8.98	83.45
MPC	40.38	14.81	1.99	15.10	6.83	14.13	84.43
SeCom	43.81	21.33	8.83	34.09	18.54	33.02	88.42
HippoRAG 2	45.62	16.66	2.91	17.01	8.27	15.93	84.88
RAPTOR	31.72	14.55	2.88	15.09	7.49	14.18	84.48
MemGAS	41.07	17.66	3.61	18.00	8.93	16.99	85.13
LightRAG	48.80	1.28	0.18	1.72	0.65	1.58	79.23
Amem	35.25	15.19	5.61	23.84	11.84	23.01	87.02
Mem0	20.09	2.23	0.21	2.48	2.51	2.34	83.80
MemTree	29.44	9.57	1.67	14.01	5.06	12.91	84.41
MemOrai (Ours)	**60.22**	**56.71**	**33.00**	**57.90**	**42.57**	**56.58**	**91.71**

and `gpt-oss-20b` for generation, ensuring a controlled comparison.

Long context and standard retrieval show limited gains: The full-history baseline achieves moderate judge scores (50.60% on LongMemEval-s), but performance drops notably on the more fragmented LOCOMO-10 (33.43%). Dense retrievers (e.g., Contriever, BGE-M3) improve lexical metrics—Contriever reaches R-1 = 32.10 on LongMemEval-s—but their GPT-4o scores remain below 42%, suggesting that embedding-based similarity alone may not reliably surface semantically relevant evidence for complex, multi-session questions.

Compression and static graphs face granularity trade-offs: Methods that compress dialogue (e.g., LLM-RSum, RAPTOR) show reduced lexical performance, possibly due to loss of fine-grained details. Graph-based approaches such as HippoRAG 2 perform well at session-level retrieval (75.53 R@3; Table 2) but exhibit substantially lower turn-level recall (27.80 R@3; Table 3), indicating that coarse structural representations may not preserve sufficient turn-level provenance for precise QA.

Embedding quality does not fully explain performance: BGE-M3, which uses its own stronger embeddings, attains the highest turn-level recall among dense retrievers (67.97 R@3; Table 3) but achieves only 47.60% GPT-4o-J (Table 1)—lower than several memory-centric methods (e.g., MemGAS: 60.20%). This suggests that high embedding quality, while helpful, may not be sufficient without mechanisms for selective retention and contextualized retrieval.

Our approach shows consistent improvements across metrics: Using the same `Contriever` embeddings as baselines, MemORAI achieves the highest recall at both granularities (90.17 R@3 session, 71.13 R@3 turn on LongMemEval-s; Tables 2, 3) and the highest GPT-4o scores (75.55% and 60.22%). Notably, it outperforms BGE-M3 in turn-level retrieval (71.13 vs. 67.97) and judge score (75.55% vs. 47.60%) despite the latter's stronger embeddings. These results suggest that the proposed components—selective memory filtering, provenance-aware graph construction, and query-adaptive ranking—may help bridge the gap between retrieval precision and generation fidelity. Further ablation studies (§4.3) examine their individual contributions.

4.3 Ablation Study

We conduct controlled ablations to assess the impact of each core component. Unless otherwise noted, all variants use the same `Contriever` embeddings and `gpt-oss-20b` backbone.

4.3.1 Selective Memory Filtering and Topic Segmentation

Table 4 evaluates the role of topic segmentation and selective memory filtering. Removing topic segmentation leads to substantial performance degradation—e.g., turn-level R@10 drops from 91.63 to 23.86 on LongMemEval-s, and from 64.68 to 27.61 on LOCOMO-10. In contrast, ablating selective filtering has a more moderate impact (e.g., −17.78 on LongMemEval-s turn R@10), suggesting that segmentation provides a stronger structural prior for memory organization.

Table 4: Topic Segmentation & Selective Memory

Method	R@10 Turn	R@10 Session
Locomo		
Current (full)	64.68	92.03
w/o Selective	57.32	87.44
w/o Topic Seg	27.61	66.02
Longmem_s		
Current (full)	91.63	98.54
w/o Selective	73.85	93.72
w/o Topic Seg	23.86	75.79

Figure 3 shows that this configuration produces significantly denser memory graphs, whereas the full pipeline yields more compact structures. This reduction in graph complexity helps suppress irrelevant connections (reducing noise during traversal) and lowers computational overhead—consistent with the observed gains in both accuracy and efficiency.

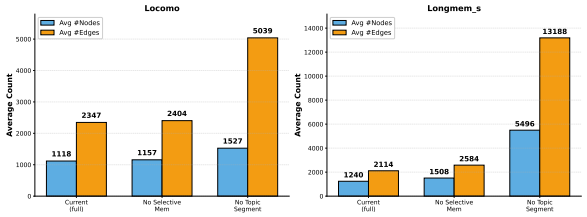

Figure 3: Graph complexity comparison across ablation configurations.

4.3.2 Dynamic Edge Weighting

Table 5 shows that dynamic edge weighting consistently improves retrieval across both benchmarks and granularities. For instance, turn-level R@10 increases by +1.88 on LongMemEval-s (89.75 → 91.63) and +2.67 on LOCOMO-10 (62.01 → 64.68) over uniform weighting. These gains—though modest in magnitude—are stable across settings, suggesting that query-conditioned edge weights help adapt retrieval to shifting dialogue context.

Table 5: Dynamic Edge Weighting

Method	R@10 Turn	R@10 Session
Locomo		
Dynamic Weight	64.68	92.03
Uniform (w=1)	62.01	91.02
Longmem_s		
Dynamic Weight	91.63	98.54
Uniform (w=1)	89.75	97.70

4.3.3 Query-Focused Subgraph Retrieval and Triplet Enrichment

We examine two design choices: (1) restricting PageRank to a query-focused subgraph, and (2) enriching retrieved turns with their supporting knowledge graph triplets.

First, Tables 6 and 7 compare full-graph versus subgraph-based retrieval. Subgraph retrieval consistently improves recall—e.g., +12.91 in turn-level R@10 on LOCOMO-10 (51.77 → 64.68)—while reducing PPR latency (14.19 ms → 12.44 ms) and maintaining high turn coverage (94.31%). On LongMemEval-s, the latency gain is larger (18.34 ms → 14.21 ms), with near-perfect coverage (99.90%). These results suggest that limiting traversal to a relevance-bounded subgraph helps suppress distant or low-signal nodes, thereby reducing noise without substantial loss of recall—particularly valuable in sparse or fragmented dialogue histories.

Table 2: Session-level retrieval performance. All methods use the same retrieval architecture. Best results in **bold**, second-best underlined.

Model	LongMemEval-s			LOCOMO-10		
	Recall@3	Recall@5	Recall@10	Recall@3	Recall@5	Recall@10
MPNet	66.17	76.38	85.11	45.92	53.98	68.58
Contriever	71.06	81.28	90.00	49.90	58.26	71.80
LLM-RSum	67.23	79.79	87.66	47.23	59.01	74.97
MPC	60.00	68.09	80.00	49.50	57.45	71.85
SeCom	71.06	80.43	89.15	53.86	62.01	73.44
HippoRAG 2	75.53	84.68	91.28	56.60	65.06	78.05
MemGAS	78.51	88.94	94.47	57.30	67.32	81.82
MemOrai (Ours)	**90.17**	**94.56**	**98.54**	**72.05**	**81.63**	**92.03**

Table 3: Turn-level retrieval performance. All methods use the same retrieval architecture. Best results in **bold**, second-best underlined.

Model	LongMemEval-s			LOCOMO-10		
	Recall@3	Recall@5	Recall@10	Recall@3	Recall@5	Recall@10
MPNet	48.25	57.58	70.53	25.36	32.40	43.15
Contriever	45.07	57.02	69.40	24.36	30.30	37.75
BGE-M3	67.97	79.36	88.14	18.03	22.63	29.52
BM25	56.72	64.68	73.69	26.05	32.10	39.21
HippoRAG 2	27.80	38.85	53.90	42.43	**52.49**	61.81
LightRAG	13.41	23.86	48.66	8.33	13.90	26.03
MemOrai (Ours)	**71.13**	**82.64**	**91.63**	**42.63**	51.97	**64.68**

Table 6: Full Graph vs Subgraph: Retrieval Performance

Method	R@10 Turn	R@10 Session
Locomo		
Full Graph	51.77	86.13
Subgraph	64.68	92.03
Longmem_s		
Full Graph	88.91	96.86
Subgraph	91.63	98.54

Table 7: Full Graph vs Subgraph: Efficiency & Coverage

Method	PPR (ms)	Turn Cov (%)
Locomo		
Full Graph	14.19	100.0
Subgraph	12.44	94.31
Longmem_s		
Full Graph	18.34	100.0
Subgraph	14.21	99.90

Second, Table 8 assesses the impact of injecting retrieved triplets during generation. Augmenting turns with their associated relational context consistently improves output quality: on LOCOMO-10, GPT-4o judge scores increase from 51.66 to 60.22 (+8.45), and BLEU more than doubles (13.58 $\rightarrow$ 33.00). Larger gains are observed on LongMemEval-s (GPT-4o-J: +13.83), where answers often require precise entity or temporal grounding. This pattern indicates that structured context helps the generator resolve ambiguities—e.g., distinguishing between similarly phrased user intents or tracking evolving preferences—yielding responses that are not only more fluent but also more factually grounded.

Table 8: Ablation Study: Impact of Triplet Context Enrichment on Generation

Method	GPT4o-J	F1	BLEU	R-L
Locomo				
Turn only	51.66	27.43	13.58	37.58
Turn + Triplets	60.22	51.89	33.00	45.97
Longmem_s				
Turn only	61.72	49.58	27.43	49.47
Turn + Triplets	75.55	56.71	33.00	56.58

5 Conclusion

We introduce **MemORAI**, a memory framework that integrates selective filtering with segment-level summarization to retain user-relevant content while

preserving global context; a provenance-enriched heterogeneous graph linking entities, turns, and segments for fine-grained, auditable retrieval; and dynamic weighted PageRank to construct query-focused subgraphs with context-aware edge weighting for prioritizing relevant evidence. Experiments on multi-session benchmarks show consistent improvements over strong baselines in turn-level recall and factual correctness under controlled conditions. These results highlight the value of jointly modeling memory granularity, temporal provenance, and query adaptation to enhance long-horizon memory utilization, enabling more coherent and reliable extended interactions.

Limitations

While MemORAI demonstrates strong performance on existing long-horizon benchmarks, its reliance on turn-level provenance and static entity linking may limit adaptability in highly dynamic or ambiguous conversational contexts—e.g., when user intent shifts abruptly or coreferences span many sessions with sparse explicit mentions. A primary challenge of our current framework remains the high computational overhead and memory requirements inherent in deploying large-scale LLMs and complex graph-based retrieval in real-time. These constraints limit accessibility in resource-constrained environments. To address this, knowledge distillation (KD) (Nguyen et al., 2026; Vuong et al., 2026; Vu et al., 2026a) has emerged as a crucial technique to transfer capabilities from powerful teacher models to more compact architectures. In future work, we plan to optimize MemORAI for more efficient deployment by integrating Small Language Models (SLMs) and specialized small-scale embedding models. We aim to leverage advanced distillation frameworks (Truong et al., 2025; Vu et al., 2026b; Le et al., 2025) to ensure that smaller models maintain high-fidelity personalized memory retrieval capabilities.

Acknowledgments

This project was supported by the Air Force Office of Scientific Research under award number FA9550-23-S-0001.

References

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

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tian Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*, 2(3).

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

Nguyen Manh Hieu, Vu Lam Anh, Hung Pham Van, Nam Le Hai, Diep Thi-Ngoc Nguyen, Linh Ngo Van, and Thien Huu Nguyen. 2025. Magix: A multi-granular adaptive graph intelligence framework for enhancing cross-lingual rag. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5202–5219.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Preprint*, arXiv:2112.09118.

Tue Le, Hoang Tran Vuong, Quyen Tran, Linh Ngo Van, Mehrtash Harandi, and Trung Le. 2025. Token-level self-play with importance-aware guidance for large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

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

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. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Bang Liu, Xinfeng Li, Jiayi Zhang, Jinlin Wang, Tanjin He, Sirui Hong, Hongzhang Liu, Shaokun Zhang,

Kaitao Song, Kunlun Zhu, Yuheng Cheng, Suyuchen Wang, Xiaoqiang Wang, Yuyu Luo, Haibo Jin, Peiyan Zhang, Ollie Liu, Jiaqi Chen, Huan Zhang, and 29 others. 2025. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *Preprint*, arXiv:2504.01990.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Preprint*, arXiv:2307.03172.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. *Preprint*, arXiv:2402.17753.

Toan Ngoc Nguyen, Nam Le Hai, Nguyen Doan Hieu, Dai An Nguyen, Linh Ngo Van, Thien Huu Nguyen, and Sang Dinh. 2025. Improving vietnamese-english cross-lingual retrieval for legal and general domains. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 142–153.

Truong Nguyen, Phi Van Dat, Ngan Nguyen, Linh Ngo Van, Trung Le, and Thanh Hong Nguyen. 2026. CTPD: cross tokenizer preference distillation. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI*, pages 37783–37790. AAAI Press.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Jianfeng Gao. 2025. On memory construction and retrieval for personalized conversational agents. *Preprint*, arXiv:2502.05589.

Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. 2024. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for llms. *arXiv preprint arXiv:2410.14052*.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. *Preprint*, arXiv:2401.18059.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Preprint*, arXiv:2004.09297.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long T. Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. 2025. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. *Preprint*, arXiv:2503.08026.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1332 others. 2025. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Joan C. Timoneda and Sebastián Vallejo Vera. 2025. Memory is all you need: Testing how model memory affects llm performance in annotation tasks. *Preprint*, arXiv:2503.04874.

Minh-Phuc Truong, Hai An Vu, Tu Vu, and Ngo Van Linh. 2025. Emo: Embedding model distillation via intra-model relation and optimal transport alignments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7605–7617.

Duc Trung Vu, Pham Khanh Chi, Dat Phi Van, Linh Ngo Van, Dinh Viet Sang, and Trung Le. 2026a. Dwa-kd: Dual-space weighting and time-warped alignment for cross-tokenizer knowledge distillation. In *Findings of the Association for Computational Linguistics: EACL*, pages 3513–3527.

Hai An Vu, Minh-Phuc Truong, Tu Vu, Nguyen Thi Ngoc Diep, Linh Ngo Van, Thien Huu Nguyen, and Trung Le. 2026b. Mol: Mixture of layers in cross-tokenizer embedding model distillation. *Knowledge-Based Systems*, 343:116001.

Hoang Tran Vuong, Tue Le, Quyen Tran, Linh Ngo Van, and Trung Le. 2026. MCW-KD: multi-cost wasserstein knowledge distillation for large language models. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI*, pages 33332–33340. AAAI Press.

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

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

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Representation Learning*, volume 2025, pages 86809–86836.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Ruifeng Yuan, Shichao Sun, Yongqi Li, Zili Wang, Ziqiang Cao, and Wenjie Li. 2024. Personalized large language model assistant with evolving conditional memory. *Preprint*, arXiv:2312.17257.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

A Dataset Statistics

Metric	LoCoMo-10
Total Conversations	10
Avg. Sessions per Conv.	27.2
Avg. Query per Conv.	198.6
Avg. Tokens per Conv.	20,078.9
Session Dates Annotated	✓
Retrieval Ground-Truth	✓
QA Ground-Truth	✓
Conversation Subject	User–User

Table 9: Statistics of the **LoCoMo-10** dataset. "Avg." denotes per-conversation averages.

Dataset Overview. LoCoMo-10 (Maharana et al., 2024) is a curated subset of the larger LoCoMo benchmark designed for evaluating long-term conversational memory. It contains ten extended *user–user dialogues*, each averaging about **27 sessions** and roughly **20k tokens**.

Unlike assistant-style datasets, LoCoMo focuses on *natural human conversation flow*, where topics evolve, reappear, and depend on long-range context. This makes it a challenging testbed for models aiming to preserve and reason over persistent memory states. Each conversation is annotated with **session timestamps**, **retrieval ground-truth**, and **QA supervision**, allowing controlled evaluation of memory construction, temporal grounding, and information recall across distant dialogue turns.

Overall, LoCoMo-10 captures the core difficulty of *multi-session coherence*—understanding entities, events, and relationships that span days or weeks of dialogue—providing a compact yet realistic benchmark for long-term memory systems like **MemORAI**.

Setup. The LoCoMo-10 benchmark divides its QA task into five reasoning categories designed to test different aspects of long-term memory: *Single-hop*, *Multi-hop*, *Temporal Reasoning*, *Open-domain Knowledge*, and *Adversarial* questions. Each type probes a distinct ability—from retrieving local facts to integrating scattered evidence across sessions or rejecting unanswerable prompts.

Results and Analysis. MemORAI achieves the highest scores on **Single-hop reasoning**, with strong margins across all metrics (**F1** = 24.67, **ROUGE-L** = 24.1, **BERTScore** = 87.32). This aligns with the system's strength in grounding on precise, session-local evidence: when the query's context resides in a single dialogue window, its description-enriched retrieval ensures that the generator accesses clean and relevant spans.

Performance remains solid on **Multi-hop** and **Temporal Reasoning** questions, indicating that adaptive propagation can recover links across sessions and maintain temporal consistency. However, scores are slightly lower (**F1** 15–17), reflecting the intrinsic difficulty of tracking multisession dependencies in user-user dialogues where events are implicit or temporally distant. In **Open-domain knowledge** cases, the model's reliance on dialogue-internal context limits factual completeness, as it does not access an external knowledge source.

The **Adversarial** subset shows the lowest scores, as expected, since these questions are intentionally unanswerable and reward the model for abstention rather than generation. **MemORAI** still maintains reasonable precision, implying partial robustness to misleading cues.

Discussion. Overall, the pattern demonstrates that **MemORAI's recall-oriented retrieval** benefits factual QA most when key information exists within reachable context windows. Tasks that demand aggregation or external world knowledge remain challenging, suggesting directions for future work. Nonetheless, LoCoMo-10 confirms that **broad and accurate coverage** remains the dominant factor in long-term conversational QA performance.

B Robustness, Efficiency, and Cost Analysis

This appendix presents supplementary experimental evidence and discussion addressing four aspects of MEMORAI: (1) robustness to backbone LLMs

of varying scales and context capacities; (2) reliability of LLM-dependent components under structured output errors; (3) indexing token cost and the trade-off between accuracy and computational expense relative to simpler retrieval approaches; and (4) graph scalability and long-term memory maintenance.

B.1 Cross-Backbone Evaluation

A potential concern is whether the empirical gains of MEMORAI are specific to a single backbone model or generalize across LLMs of different scales and native long-context capacities. To address this, we expand our evaluation to three open-source backbones covering a substantially broader range of scales and supported context lengths: `Qwen3-8B` (32,768 context length), `openai/gpt-oss-20B` (131,072 context length), and `Qwen3-30B-A3B` (262,144 context length). Retrieval and generation performance on the LOCOMO benchmark are reported in Tables 12 and 13.

The results demonstrate consistent effectiveness across all tested backbones. Notably, even the smallest backbone (`Qwen3-8B`) achieves competitive performance and continues to outperform strong baselines, confirming that the gains are not attributable to any single large model. Furthermore, as backbone scale and context capacity increase, MEMORAI yields further improvements across both retrieval and generation metrics, indicating that the framework scales well with stronger long-context LLMs rather than being undermined by their native capabilities. These results support the conclusion that observed improvements arise from the framework design itself—selective memory filtering, provenance-enriched graph construction, and query-adaptive retrieval—rather than from a particular backbone choice.

B.2 Robustness to Structured Output Errors

Several components of MEMORAI, including selective memory filtering and knowledge graph extraction, rely on LLM prompting to produce structured outputs (e.g., JSON or schema-compliant extractions). A legitimate concern is whether errors in these outputs—such as malformed JSON or schema violations—could degrade graph quality and downstream retrieval performance. To examine this systematically, we measure the *structured output error rate*: the proportion of LLM outputs that fail to produce valid, schema-compliant structured responses. Results are reported across all compared methods and both benchmarks in Table 14.

MEMORAI maintains consistently low error rates across both benchmarks, demonstrating stable structured extraction behavior in practice. By contrast, A-Mem exhibits error rates exceeding 73%, substantially undermining the integrity of its knowledge graph and subsequent retrieval. The majority of observed failures across methods arise from malformed JSON outputs—specifically missing closing brackets or unescaped quotation marks—that prevent extracted knowledge from being parsed correctly.

The low error rate of MEMORAI is attributable to two design decisions: (i) extraction prompts are engineered with explicit schema constraints and formatting instructions tailored to triplet and provenance extraction; and (ii) a schema validation step during the offline indexing phase discards ill-formed outputs before they propagate into the knowledge graph. Together, these measures substantially reduce formatting errors and yield reliable schema-compliant outputs, confirming that the LLM-dependent components of MEMORAI are robust to structured extraction failures under realistic operating conditions. The efficacy of leveraging LLMs for accurately extracting multi-aspect structured information from complex text has also been validated in recent domain-specific retrieval pipelines (Nguyen et al., 2025).

B.3 Indexing Token Cost and Comparison with Simpler Retrieval Approaches

Token Usage Per Session. MEMORAI is designed to support relational and provenance-aware memory, which inherently requires more structured processing than lightweight embedding-only pipelines. Constructing compressed memory units and entity-level links involves LLM-based extraction beyond simple similarity indexing, introducing higher token consumption during the indexing phase. We acknowledge this trade-off and report token usage statistics per session for transparency in Table 15.

Importantly, the indexing phase in MEMORAI is performed as an *offline or asynchronous update process*, decoupled from real-time response generation. Consequently, indexing cost does not affect response latency at inference time, as retrieval operates over an already-constructed memory graph. This design is consistent with prior graph-based memory systems (Gutiérrez et al., 2025). Fur-

Table 10: Session-level retrieval performance. LME = LongMemEval, LOCO = LOCOMO-10.

Model	LME R@3	LME R@5	LME R@10	LOCO R@3	LOCO R@5	LOCO R@10
Contriever	71.06	81.28	90.00	49.90	58.26	71.80
HippoRAG 2	75.53	84.68	91.28	56.60	65.06	78.05
MEMORAI	**90.17**	**94.56**	**98.54**	**72.05**	**81.63**	**92.03**

Table 11: Turn-level retrieval performance.

Model	LME R@3	LME R@5	LME R@10	LOCO R@3	LOCO R@5	LOCO R@10
Contriever	45.07	57.02	69.40	24.36	30.30	37.75
HippoRAG 2	27.80	38.85	53.90	42.43	52.49	61.81
MEMORAI	**71.13**	**82.64**	**91.63**	**42.63**	**51.97**	**64.68**

thermore, MEMORAI does not rely on extremely large proprietary models: our system uses the open-source `openai/gpt-oss-20B` backbone (3.6B active parameters), which is substantially more accessible than systems requiring GPT-4o for memory construction. While MEMORAI incurs higher indexing cost than lightweight methods such as Mem0-g, it remains notably more token-efficient than other graph-heavy approaches (e.g., A-Mem), while delivering over 4% absolute improvement in retrieval performance on LOCOMO. Managing these computational trade-offs is a pervasive challenge in LLM deployment, akin to balancing multi-cost alignments in cross-tokenizer knowledge distillation (Vuong et al., 2026).

Why Graph-Based Retrieval over Embedding-Based Approaches? Graph-based methods explicitly model relationships between entities and events, enabling meaningful connections even when related facts do not directly co-occur in the same context. This structure natively supports multi-hop reasoning and provenance-aware retrieval across long conversation horizons. By contrast, embedding-only approaches rely primarily on similarity signals and do not capture relational dependencies, making complex reasoning less reliable (Gutiérrez et al., 2025).

This advantage is confirmed in our controlled experiment using the same embedding backbone (Contriever), where graph-based retrieval yields large gains at both session and turn levels (Tables 10 and 11), confirming that the improvements come from relational modeling rather than stronger embeddings alone.

B.4 Efficiency of Dynamic Weighted PageRank

The Dynamic Weighted PageRank (DWPR) mechanism modifies standard Personalized PageRank (PPR) by incorporating query-conditioned edge weights to improve ranking quality for semantically relevant but sparsely connected nodes. We examine whether this modification introduces meaningful computational overhead and whether its retrieval gains justify the added complexity over uniform-weight PPR.

Latency Comparison. Table 16 compares end-to-end retrieval latency (in milliseconds) for Traditional PPR and DWPR on both benchmarks.

The additional latency introduced by DWPR is minimal—approximately 4.5 ms on LOCOMO and 13.6 ms on LongMemEval (less than 1.4% and 1.1% overhead, respectively)—confirming that DWPR operates at effectively the same computational cost as standard PPR.

Retrieval Improvement. Despite its negligible overhead, DWPR yields consistent improvements across both benchmarks and retrieval granularities, as shown in Table 16. The purpose of DWPR is not to replace PPR with a heavier algorithm, but to introduce a lightweight modification that improves ranking quality while preserving the efficiency of standard PPR. These results establish a favorable cost-effectiveness profile, particularly in long-context settings where modest retrieval improvements translate into meaningful downstream generation gains.

B.5 Graph Scalability and Long-Term Memory Maintenance

A natural question for any continuously operating memory system concerns long-term graph scala-

Table 12: Retrieval performance of MEMORAI on LOCOMO across backbone models of increasing scale and context capacity.

Model	Turn R@3	Turn R@5	Turn R@10	Session R@3	Session R@5	Session R@10
Qwen3-8B	41.00	52.22	63.51	64.83	75.18	87.08
openai/gpt-oss-20B	42.63	51.97	64.68	72.05	81.63	92.03
Qwen3-30B-A3B	**55.34**	**64.12**	**74.13**	**77.55**	**84.81**	**92.28**

Table 13: Generation performance of MEMORAI on LOCOMO across backbone models. Metrics: GPT-4o-J F1, BLEU, ROUGE-1/2/L, and BERTScore.

Model	GPT-4o-J F1	BLEU	R-1	R-2	R-L	BERTScore
Qwen3-8B	50.65	45.58	29.66	47.15	35.02	89.52
openai/gpt-oss-20B	60.22	56.71	33.00	57.90	42.57	91.71
Qwen3-30B-A3B	**63.26**	**57.82**	**34.97**	**58.88**	**44.58**	**91.80**

Table 14: Structured output error rate (%) on LOCOMO and LongMemEval. Secom and MemGas do not rely on JSON-based structured extraction (n/a).

Method	LOCOMO	LongMemEval
A-Mem	73.91	74.07
Mem0-g	2.43	9.10
Secom	n/a	n/a
MemGas	n/a	n/a
Mem0	10.39	7.60
MEMORAI	**3.70**	**2.10**

Table 15: Token usage per session (Indexing + Updates) for all methods on LOCOMO and LongMemEval.

	LOCOMO		LongMemEval	
Method	Input Tok.	Output Tok.	Input Tok.	Output Tok.
A-Mem	12065.1	10771.6	19330.0	5056.3
Mem0-g	783.0	1298.0	889.6	1442.7
Secom	1633.6	802.2	3331.9	905.7
MemGas	1795.9	549.0	5236.1	965.9
Mem0	994.6	1012.4	1174.7	1344.9
MEMORAI	9700.4	3627.8	13859.5	2230.1

Table 16: Latency and retrieval improvement of Dynamic Weighted PageRank vs. Traditional PPR.

Benchmark	Method	Latency (ms)	R@10 Turn	R@10 Session
LOCOMO	Traditional PPR	345.46	62.01	91.02
	DW PageRank	349.97	64.68	92.03
LongMemEval	Traditional PPR	1256.53	89.75	97.70
	DW PageRank	1270.12	91.63	98.54

bility: while Selective Filtering reduces low-utility input at the ingestion stage, the knowledge graph itself grows under extended deployment. We address this concern along two dimensions.

Structural Scalability via Incremental Updates. A key architectural advantage of MEMORAI is its support for fully incremental graph updates. Unlike tree-based retrieval approaches such as RAPTOR (Sarthi et al., 2024), which require recursive summarization over raw text and must rebuild the entire tree structure whenever new information is added, our method appends new information as nodes and edges without requiring any re-encoding or reconstruction of the existing memory store. This design closely mirrors LightRAG (Guo et al., 2024), which similarly adopts a modular graph structure for efficient, localized memory updates at scale. As a result, MEMORAI is inherently well-suited to continual, open-ended conversational settings where new information arrives continuously and unpredictably.

C Prompt Templates

C.1 Conversation Segmentation

```
You are an expert in conversational discourse analysis. Your task is to perform topic segmentation
on a multi-turn dialogue.

TASK DEFINITION
Segment the conversation into topically coherent units based on semantic relatedness. Successive
conversational turns discussing the same topic should be grouped into the same segment. Create
new segments when topic shifts occur.

INPUT FORMAT
A numbered sequence of conversational turns, where each message is labeled with its speaker role:
- Message i: [speaker]: [utterance]

OUTPUT FORMAT
Return ONLY a JSON list of lists containing integer indices: [[seg1_indices], [seg2_indices], ...]
- Each inner list represents one topical segment
- Indices must be integers corresponding to message numbers
- Segments should be non-overlapping and cover all messages
- Maintain chronological order

SEGMENTATION CRITERIA
- Semantic coherence: Messages within a segment should share topical focus
- Topic shift detection: Create new segments when conversation shifts to distinct topics
- Contextual continuity: Consider discourse markers and referential dependencies

EXAMPLE
INPUT CONVERSATION:
Message 0: user: Hey, how are you?
Message 1: assistant: I'm good, thanks! Just finishing up some work.
Message 2: user: Speaking of work, did you see the email about the project deadline?
Message 3: assistant: Yeah, it's been moved to next Friday.
Message 4: user: Okay, that gives us more time. I'll update the project plan.
Message 5: assistant: Perfect, thanks.
Message 6: user: On a different note, are you free this weekend? I was thinking of going hiking.
Message 7: assistant: Oh, that sounds great! I'm free on Saturday.

REQUIRED OUTPUT:

[[0,1,2,3,4,5],[6,7]]
EXPLANATION: Messages 0–5 form one segment discussing work-related topics. Message 6 signals a
topic shift to weekend plans, forming a new segment with message 7.

–

INPUT CONVERSATION
{numbered_messages_str}

REQUIRED OUTPUT
Return only the JSON list of segment indices below:
```

Figure 4: Conversation Segmentation

C.2 Selective Memory Filtering

```
You are an expert Data Curator who builds user profiles from conversations.

TASK
Analyze the conversation and return a JSON array of message indices that contain valuable
information about the user.

WHAT COUNTS AS "VALUABLE USER INFORMATION"
Keep only messages that help identify the user's personal context, including:
1) Personal information
- Biographical facts: name, age, job, education, location
- Possessions/ownership, experiences, achievements
- Relationships, life events, specific details they shared

2) Interests / preferences / goals
- Likes/dislikes, habits, goals
- Requests for recommendations or advice that reveal real needs
- Questions that reveal their situation

3) Contextual exchanges
- Questions that clarify the user's specific context
- Personalized suggestions the user requested
- Responses that explicitly reference details the user mentioned earlier

WHAT TO SKIP
- Generic knowledge not tied to this user
- General definitions or instructions applicable to anyone
- Content with no connection to the user's personal context

OUTPUT FORMAT
Return ONLY a JSON array of message indices to keep. Do not add any text outside the JSON array.

EXAMPLES

Example 1
[0] user: What's photosynthesis?
[1] assistant: Photosynthesis is the process where plants convert sunlight into energy using
chlorophyll.
Output: [0]
Why keep: Message [0] indicates the user's learning interest/need. Message [1] is generic
knowledge and does not add user-specific profile information.

Example 2
[0] user: My cat Luna keeps scratching the furniture
[1] assistant: Since Luna is scratching the furniture, try placing a scratching post near her
favorite spots.
Output: [0, 1]
Why keep: Message [0] contains ownership and a specific personal detail (a cat named Luna) plus a
concrete problem. Message [1] is personalized to the user's stated context.

Example 3
[0] Tom: Alex, are you moving to Berlin next week?
[1] Alex: Yeah. I'm moving to Berlin because I got a data engineer job. I'm worried about rent
because my budget is only around 1,200 EUR/month.
[2] Tom: Are you going alone or with someone?
[3] Alex: Alone. I prefer a place near the U-Bahn so commuting is easy.
Output: [1, 3]
Why keep: [1] includes location (Berlin), job (data engineer), and a constraint/goal (rent budget).
[3] adds living situation (alone) and a preference (near U-Bahn).

Conversation:
{formatted_conv}
Output:
```

Figure 5: Selective Memory Filtering

C.3 Segment Summarization

```
Summarize the following conversation segment into a concise summary (2–3 sentences).

Focus on:
- Main topic discussed
- Key information exchanged
- Important facts or decisions

Segment:
{segment_content}

Summary:
```

Figure 6: Segment Summarization

C.4 Entity Description Extraction

```
Based on the following conversation segment, provide a brief description for each entity in context.

IMPORTANT: For each description, cite the TURN INDICES (not message indices) where the information comes from.

Segment:
{segment}

Entities to describe: {entity_list}

For each entity, write a 1–2 sentence description that captures what we learn about it in this segment, and cite the turn indices used.

Output format (one entity per line):
entity | description | turn_index1,turn_index2,...

Example output:
User | A software engineer leading a team of 8 at Google, working on a search ranking algorithm project with a tight deadline | 1
Google | The company where the user works, currently launching a new project | 1

Generate descriptions:
```

Figure 7: Entity Description Extraction

C.5 Answer Generation prompt

```
Based on the provided conversation context and timestamps, answer the following question by adhering to these strict rules:
1. Precision: Provide the short possible answer (short phrase or single value). Use words from the context whenever possible.
2. Verification: First verify if the premise of the question matches the information in the context. If the specific detail is not mentioned or cannot be determined, strictly answer: 'The information provided is not enough'.
3. Recency: If facts conflict or change over time, rely on the most recent information provided by the user. Ignore outdated facts.
4. Temporal Reasoning: If the question involves dates or durations, calculate them accurately using the provided conversation timestamps.
5. Source Attribution: If the question asks specifically about what the Assistant or User said, quote their exact words from the conversation.
Based on the following context, answer the question.
{context}
Question: {query}
Answer:"""
```

Figure 8: Answer Generation prompt

C.6 Triplet Extraction with Provenance

```
You are a knowledge graph extractor that identifies factual statements from conversation
participants.

CORE PRINCIPLES:
1. Extract explicitly stated information only — avoid inference
2. Focus on all conversation participants equally
3. Capture stated facts, preferences, interests, and plans

EXTRACTION RULES:
1. Equal Treatment: Extract factual statements from any participant
2. Speaker Identification: Use the participant's identifier (username, role label, or "Speaker[N]")
3. Pronoun Resolution: Replace pronouns with the speaker's identifier
4. Multi-turn Tracking: If information spans multiple messages, record all relevant indices

RELATIONSHIP TYPES:
- Identity: is, is a, has age, is from, lives in
- Professional: works at, studies at, has role
- Preferences: likes, prefers, enjoys, is interested in
- Intentions: is planning to, wants to, considering

OUTPUT FORMAT:
entity1|relation|entity2|message_indices

EXAMPLES:

Example 1:
Message 0: An: I'm a designer at Apple.
Message 1: Binh: I work at Microsoft as a PM.
Message 2: An: I love cross-company projects.
Output:
An|has role|designer|0
An|works at|Apple|0
Binh|works at|Microsoft|1
Binh|has role|PM|1
An|likes|cross-company projects|2

Example 2:
Message 0: Sam: I live in Tokyo.
Message 1: Assistant: Interesting! Do you work there too?
Message 2: Sam: Yes, I've been living and working in Tokyo for 2 years.
Output:
Sam|lives in|Tokyo|0,2
Sam|works in|Tokyo|2
Sam|has lived in Tokyo for|2 years|2

CONVERSATION TO ANALYZE:
{segment_text}

Extract all factual triplets, one per line:
```

Figure 9: Triplet Extraction with Provenance

C.7 GPT-4 Judge Prompt

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


[User Question]
{question}


[The Start of Reference Answer]
{answer}
[The End of Reference Answer]

[The Start of Model's Response]
{response}
[The End of Model's Response]

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

Figure 10: GPT-4 Judge Prompt