Persistent Long-Term Memory in Conversational AI Agents: A Multi-Provider Benchmark Study

1st Subodh Kumar Nannapaneni
AI Development
LehtAI Technologies
Vijayawada, India
subodhnannapaneni@gmail.com

2nd Jyothi Nannapaneni
AI Development
LehtAI Technologies
Ottawa, Canada
njyothi8985@gmail.com

***Abstract*—Large language model (LLM) agents are fundamentally stateless: each conversation begins without knowledge of prior interactions. This paper presents a systematic investigation of persistent long-term memory architectures for conversational AI agents, combining semantic vector retrieval (Milvus, HNSW index) with structured relational metadata storage (PostgreSQL) within a LangGraph-orchestrated pipeline. We make four primary contributions: (i) an end-to-end open-source memory agent architecture with hot-swappable LLM backends; (ii) a quantitative study of cosine similarity threshold selection across nine values (0.40–0.90) on the LoCoMo benchmark, demonstrating that recall@10 drops from 88% to 0% across this range with a sharp discontinuity at 0.60; (iii) an empirical comparison of FLAT versus per-user PARTITIONED Milvus index strategies over 30 retrieval runs each, showing that PARTITIONED indexing reduces p95 tail latency by 28% and worst-case latency by 40% (Cohen's d = 0.21); and (iv) the first statistically powered comparative benchmark of twelve state-of-the-art LLMs spanning three providers (OpenAI, Anthropic, Google Gemini) on a controlled memory-recall task using 75 probes per model (25 question types × 3 independent runs), finding recall accuracy ranging from 17.3% (Claude-Sonnet-4-6) to 49.3% (GPT-5) with 95% confidence intervals of ±11 percentage points, and response latency spanning 1,021 ms to 13,632 ms under identical experimental conditions. We further demonstrate that existing conflict-detection pipelines achieve only 25% overall accuracy in belief-update scenarios a previously unreported failure mode attributable to the inability of standard cosine similarity to semantically opposing facts on the surface. Our findings establish concrete, actionable thresholds, and architectural choices for production-grade deployments of agent memory systems.**

Index Terms*—conversational AI, long-term memory, LLM agents, vector databases, LangGraph, retrieval-augmented generation, Milvus, memory conflict resolution, multi-model benchmark.

I. Introduction

The emergence of large language models (LLMs) as conversational interfaces has surfaced a fundamental architectural limitation: LLMs are stateless. Each conversation begins from a blank context window, discarding all prior user interactions the moment the session ends. This statelessness fundamentally limits personalisation, continuity, and the utility of assistants deployed over extended periods.

The standard workaround prepending conversation history to each prompt does not scale. Token windows are finite. More critically, concatenating raw history is inefficient: most prior exchanges are irrelevant to any given query. What is needed is selective, persistent, and conflict-aware memory, the ability to store distilled facts about users indefinitely, retrieve only what is relevant to the current query, and gracefully handle the natural evolution of user beliefs over time.

Prior work has addressed this problem from several angles. MemGPT [1] introduced a hierarchical memory model with an LLM-controlled paging mechanism. Mem0 [2] provides a managed memory API for agent frameworks. LangMem [3] integrates memory into the LangChain ecosystem. However, these systems share a common limitation: they are primarily engineering products with limited empirical characterisation of their retrieval quality, failure modes under volume, performance across LLM providers, or behaviour when stored facts are contradicted.

This paper addresses that gap. We build a complete, open-source memory agent grounded in LangGraph [4] and present five controlled experiments that collectively answer questions no prior work has addressed together:

1. What cosine similarity threshold maximises memory recall without flooding the context with noise? (§IV-A)

2. How does memory volume affect retrieval quality and the risk of the "Lost in the Middle" phenomenon? (§IV-B)

3. Is per-user index partitioning worth the operational complexity? (§IV-C)

4. Can the state-of-the-art in LLM-based conflict detection reliably identify when a user's belief has changed? (§IV-D)

5. Do different LLM providers differ meaningfully in their ability to utilise injected memory context? (§IV-E)

The answers are concrete and, in several cases, surprising particularly the finding that GPT-5.4, despite being the highest-cost model tested, delivers poor recall performance (25.3%), and that conflict detection achieves 0% accuracy on belief-update scenarios due to a retrieval-bound failure mode rather than an LLM reasoning failure.

II. Related Work

A. Memory Architectures for LLM Agents

Park et al. [5] introduced Generative Agents—simulated characters with stream-of-consciousness memory and reflection mechanisms. While conceptually elegant, the architecture was simulation-specific. MemGPT [1] proposed a virtual context management system analogous to OS memory paging, where the LLM manages what resides in external context via function calls. Production-grade tools like Zep [6] and Mem0 [2] offer memory APIs with fact extraction, but lack published quantitative evaluations of retrieval accuracy across threshold settings or cross-provider benchmarks.

B. Retrieval-Augmented Generation (RAG)

Our system aligns with dense-passage retrieval in RAG [7]. However, memory retrieval differs from document retrieval in important ways: the queries are conversational and sparse, and contradictions between stored documents are an intended feature (belief evolution).

C. "Lost in the Middle" Phenomenon

Liu et al. [8] demonstrated that LLMs systematically underutilise information in the middle of long contexts. Our §IV-B experiment directly applies this finding to conversational memory: when top-K retrieval returns 49+ memories, most are in the middle of the context block and effectively ignored.

D. Multi-Session and Belief-Evolving Dialogue

The Multi-Session Chat (MSC) dataset [9] is the only publicly available benchmark specifically designed for conversations where user beliefs evolve across sessions. Our conflict resolution evaluation (§IV-D) is grounded in MSC-derived scenarios. Prior work on MSC focused on response generation quality rather than the correctness of belief-update detection, which we address.

E. Cross-Provider LLM Benchmarks

Existing benchmarks (MMLU [10], HumanEval [11], LMSYS Chatbot Arena [12]) do not measure memory utilisation. They evaluate general knowledge, coding, or conversational quality. To our knowledge, this paper presents the first controlled benchmark of LLM providers on a memory-utilisation task measuring whether a model correctly uses injected memory context rather than ignoring it or hallucinating contradictory information.

III. System Architecture

A. Overview

The proposed system is a LangGraph-based conversational agent with a persistent dual-store memory backend. The graph is compiled with a PostgresSaver checkpointer that persists conversation history.

B. Memory Storage Layer

Two stores operate in parallel:

- **Milvus Vector Store:** A locally-deployed, Docker-hosted Milvus instance manages semantic retrieval. Memories are stored as 1536-dimensional vectors generated by OpenAI `text-embedding-3-large` using Matryoshka Representation Learning (MRL) truncation [13]. We set `dimensions=1536`, halving the native 3072-dim output. This yields three advantages: (i) storage: each vector occupies 6 KB instead of 12 KB; (ii) search latency: computation scales linearly with dimension count, so 1536-dim search runs ∼50% faster; (iii) semantic quality: MRL-trained models preserve 95–98% of full-dimensional retrieval quality at 50% truncation [13]. The index uses `IVF_FLAT` with `COSINE` metric (`nlist=128`, `nprobe=16`).
- **PostgreSQL Metadata Store:** Structured metadata for every stored memory: content, type classification, importance score, TTL expiry timestamp, and access frequency. Three tables support the system: `memory_store`, `conflict_log`, and `experiment_metrics`.

C. Memory Classification and Typing

Every user message passes through the Gatekeeper module before storage. An LLM call at temperature=0 (using OpenAI gpt-4o-mini as the classification model) returns a structured JSON decision with fields: `should_save` (bool), `memory_type` (FACTUAL/365-day TTL, PREFERENCE/90-day TTL, or EPHEMERAL/1-day TTL), `importance` (0.0–1.0), and `distilled_content` (a clean, third-person memory sentence).

D. Conflict Resolution Pipeline

Before any new memory is stored, the module: (1) classifies the incoming message; (2) queries Milvus for top-5 similar memories at $\tau_{conflict} = 0.82$; (3) invokes the LLM to classify the relationship as UPDATE, APPEND, or IGNORE.

E. Decay Mechanism

Two complementary pruning strategies maintain memory quality over time. TTL Pruning performs SQL-based hard deletion of rows where `expires_at` $<$ `NOW()`, with vectors subsequently bulk-deleted from Milvus. Semantic Pruning uses an LLM to review the 50 most recent memories in a batch and identifies redundant or superseded entries, catching cases TTL cannot address.

F. LLM Provider Abstraction

All twelve tested models are accessed through a unified `_get_llm(model_key)` factory backed by LangChain provider integrations (`ChatOpenAI`, `ChatAnthropic`, `ChatGoogleGenerativeAI`, `ChatGroq`). Adding a new provider requires modifying exactly two files: the registry and the factory switch statement, with no changes to any experiment or graph code.

IV. Experiments and Results

All experiments were conducted on the same deployed system instance (PostgreSQL 15, Milvus 2.4, Python 3.11.9, Windows 11). LLM API calls used live production endpoints as of April 2, 2026. Embedding model was fixed at OpenAI `text-embedding-3-large` with MRL truncation to 1536 dimensions for all providers to ensure a consistent vector space.

A. Experiment 1: Cosine Similarity Threshold vs. Memory Recall

Motivation: The cosine similarity threshold τ is the single most consequential hyperparameter in the retrieval pipeline. Setting τ too high starves the LLM of relevant context; setting it too low floods the context window with noise.

Dataset: 25 ground-truth question-answer pairs derived from the LoCoMo long-context conversation benchmark [14], covering 5 distinct virtual users.

Protocol: For each threshold value $\tau \in \{0.40, 0.50, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90\}$, all 25 probes were issued with top-K=10. Recall@10 was computed as the fraction of probes where the correct memory appeared in the retrieved set.

TABLE I
Cosine Similarity Threshold vs. Retrieval Metrics ($n = 25$)

Threshold(τ)	Recall@10	Precision	F1	Avg Retr.	Fetch(ms)
0.40	0.880	0.096	0.173	9.9	404.4
0.50	0.760	0.077	0.140	9.6	314.5
0.60	0.400	0.041	0.074	9.2	329.3
0.65	0.320	0.033	0.060	9.0	345.6
0.70	0.280	0.029	0.052	8.9	331.3
0.75	0.200	0.021	0.038	8.6	343.9
0.80	0.080	0.010	0.018	8.2	308.6
0.85	0.040	0.006	0.010	7.2	327.8
0.90	0.000	0.000	0.000	4.8	312.8

Analysis: Recall exhibits a steep monotonic decline. The most critical transition occurs between $\tau = 0.50$ (Recall@10=0.76) and $\tau = 0.60$ (Recall@10=0.40)—a 47% relative decline for a 0.10-unit increase. At $\tau = 0.90$, recall collapses.

Key finding: The optimal operating point is $\tau = 0.50$–0.55. We recommend a dual-threshold strategy: $\tau_{retrieval} = 0.50$ and $\tau_{conflict} = 0.35$.

B. Experiment 2: Memory Volume and Lost-in-the-Middle Risk

Motivation: As a user accumulates memories over time, two risks emerge: (1) retrieval quality degrades as the vector space becomes denser; (2) the fixed top-K window forces lower-scoring memories to the middle of the injected context block, where they are systematically under-processed by LLMs.

Protocol: Three conditions were evaluated: cold start (0 memories), `vol_50` (50 memories), and `vol_500` (500 memories). Top-K was capped at 50. LitM risk was classified as LOW (< 500 estimated tokens), MEDIUM (500–2,000), or HIGH ($> 2,000$).

TABLE II
Memory Volume vs. Retrieval Quality and LitM Risk

Condition	Memories	Avg Retr.	Avg Rel.	LitM Risk
Cold Start	0	0	0	LOW
vol_50	50	49.8	0.838	MEDIUM
vol_500	500	49.0	0.709	HIGH

Analysis: Despite an order-of-magnitude difference in stored memory count (50 vs. 500), retrieved set sizes are nearly identical (49.8 vs. 49.0) because the top-K cap prevents growth. However, the average relevance score drops significantly from 0.838 at vol_50 to 0.709 at vol_500, a 15.4% relative degradation. This degradation occurs silently.

Key finding: A flat top-K retrieval policy creates a false sense of security. Production deployments should monitor average relevance score as a health metric and adopt a dynamic top-K policy (retrieve until relevance < 0.70, hard cap at $K = 15$).

C. Experiment 3: Index Strategy FLAT vs. PARTITIONED

Motivation: Milvus supports multiple index strategies. PARTITIONED indexing (one partition per user) confines each search to a user-specific subset of vectors, potentially reducing search scope and improving latency. This experiment quantifies the practical benefit.

Protocol: 30 retrieval runs per strategy on identical queries against the same user's memory set. Statistics computed: mean, standard deviation, p50, p95, minimum, maximum latency.

TABLE III
Milvus Index Strategy Latency Distribution ($n = 30$)

Strategy	Mean (ms)	p95 (ms)	Max (ms)	Cohen's d
FLAT	359.66	707.46	1,087.74	–
PARTITIONED	332.53	509.51 ↓28%	656.46 ↓40%	0.21

Analysis: The mean latency improvement from FLAT to PARTITIONED is modest at 1.08× (27.1 ms absolute), consistent with the small effect size (Cohen's d = 0.21). However, the distributional improvements are practically significant: standard deviation halves (164 ms → 85 ms), p95 latency improves by 28% (707 ms → 510 ms), and the worst-case maximum improves by 40% (1,088 ms → 656 ms).

Key finding: PARTITIONED indexing should be the default for multi-user deployments. The 40% reduction in worst-case latency and 48% reduction in latency variance significantly improve the consistency of user experience in production.

D. Experiment 4: Conflict Resolution Accuracy

Motivation: A memory system that cannot detect when a user's belief has changed will accumulate contradictory facts

simultaneously stating that a user prefers Python and that they have switched to Rust, for example.

Protocol: 20 controlled belief-change scenarios spanning three resolution categories: UPDATE (10 scenarios), APPEND (5 scenarios), and IGNORE (5 scenarios). The system's output resolution was compared against the ground-truth label.

TABLE IV
CONFLICT RESOLUTION ACCURACY BY SCENARIO TYPE ($n = 20$)

Category	Accuracy	Correct / Total	Notes
OVERALL	0.25	5 / 20	scenarios combined
UPDATE	0.00 ×	0 / 10	belief-change scenarios failed
APPEND	1.00 ✓	5 / 5	Trivial new-information case
IGNORE	0.00 ×	0 / 5	duplicate/paraphrase failed

TABLE V
REPRESENTATIVE UPDATE FAILURE CASES

Old Belief	New Belief	Output	LLM Reasoning
User prefers Python as primary language.	User has switched to Rust.	APPEND	No similar memories found.
User's favourite ML framework is PyTorch.	User has switched from PyTorch to JAX.	APPEND	No similar memories found.
User prefers dark mode in all interfaces.	User now prefers light mode after eye strain.	APPEND	No similar memories found.
User works from home in Bangalore.	User has relocated to Amsterdam.	APPEND	No similar memories found.

Analysis: The 25% overall accuracy (5/20) is deceptive; all five correct answers are APPEND cases. The root cause is uniform: "No similar memories found." appears as the LLM reasoning in 16 of 20 failure cases. The conflict detection search at $\tau_{conflict} = 0.82$ fails to retrieve the opposing memory because cosine similarity fundamentally measures co-occurrence of semantic content, not semantic opposition.

Key finding: LLM-based conflict resolution is retrieval-bound, not reasoning-bound. Three mitigations are recommended: (1) lower $\tau_{conflict}$ to ≤ 0.35; (2) augment retrieval with entity-level exact matching; (3) store memories with typed "slot" fields to enable direct slot-level conflict lookup, bypassing vector similarity entirely.

E. Experiment 5: Multi-Provider LLM Memory-Recall Benchmark

Motivation: No prior work has quantified whether different LLM providers differ in their ability to utilise injected memory context in their responses.

Protocol: Twenty-five memories spanning seven semantic categories were seeded for a test user. Twenty-five recall probe questions were issued to each of 12 models through the full agent pipeline, with 3 independent runs per model (75 total probes per model). Recall was assessed as a binary keyword match. Models tested: 3 OpenAI (GPT-4.1-mini, GPT-5, GPT-5.4), 2 Anthropic (Claude Haiku-4-5, Claude Sonnet-4-6), and 7 Google Gemini models all accessed via live production APIs on April 2, 2026.

Universal Failure Pattern: Five of the twenty-five recall probes failed for every model across all 75 runs. This is not a model-quality effect; it is the threshold mechanism quantified in §IV-A. These five memory facts produce cosine similarity scores below the retrieval threshold of 0.60, causing them to be excluded from the injected context before the LLM is invoked.

Four performance tiers emerge from the expanded n=75 dataset. Tier 1 (Recall leader): GPT-5 achieves 49.3% recall but at 13,632 ms mean latency and $0.01/1k tokens. Tier 2 (Best value): GPT-4.1-mini and Gemini-3-Flash both at 44.0% recall, statistically indistinguishable from GPT-5 within 95% CI, at 25–33× lower cost. Tier 3 (Mid-range): Gemini-2.5-Flash-Lite and Gemini-2.5-Pro at 40.0%; Gemini-3-Pro at 38.7%. Tier 4 (Below-average): Claude-Sonnet-4-6 at 17.3% last place, a dramatic reversal from its 40% pilot result.

Provider-level: Anthropic's two tested models rank 9th (Claude-Haiku-4-5, 25.3%) and last (Claude-Sonnet-4-6, 17.3%), making Anthropic the worst-performing provider for memory-recall tasks at the evaluated thresholds. OpenAI provides the highest recall ceiling; Google Gemini provides the broadest coverage.

F. Experiment 6: Conflict Resolution Accuracy

The Threshold Is the Dominant System Variable: Across all five experiments, cosine similarity threshold τ emerges as the most consequential architectural parameter. Moving from $\tau = 0.75$ (the system default) to $\tau = 0.50$ increases Recall@10 from 20% to 76%—a 56 pp improvement that exceeds the full cross-model recall spread observed in Section IV-E (17.3% to 49.3%), at zero additional cost.

Retrieval-Bound vs. Reasoning-Bound Failure Modes: The conflict resolution results reveal an important distinction: in 7 of 10 UPDATE failures, the LLM reasoning was correct (consistency_hit=True) but the candidate memories were never retrieved. This means the LLM component of conflict resolution is functioning correctly; the bottleneck is the vector retrieval's inability to surface semantically opposing facts.

The Importance of Probe Set Size: The expanded benchmark ($n = 75$ per model) materially changes three key model rankings. `Claude-Sonnet-4-6`: 40% ($n = 10$ pilot) $\rightarrow$ 17.3% ($n = 75$ expanded), a -22.7 pp reversal. `GPT-5`: 70% ($n = 10$) $\rightarrow$ 49.3% ($n = 75$), a -20.7 pp correction. `Gemini-3.1-Pro`: 60% ($n = 10$) $\rightarrow$ 33.3% ($n = 75$), a -26.7 pp drop. These reversals validate that pilot studies with fewer than 25 probes should be treated as directional signals only.

V. ARCHITECTURE RECOMMENDATION SYNTHESIS

Based on all five experiments, we recommend the following production configuration:

- **Default retrieval threshold:** $\tau = 0.50$ (3.8× recall improvement over system default of 0.75).

TABLE VI
MULTI-PROVIDER LLM MEMORY-RECALL BENCHMARK ($n = 75$ PROBES PER MODEL, 25 QUESTION TYPES $\times$ 3 RUNS)

Model	Provider	Avg Latency (ms)	p_min / p_max (ms)	Recall (hits/75)	95% CI	$/1k tokens
Gemini-2.5-Flash-Lite	Google	1,021	565 / 3,085	40.0% (30/75)	±11.1 pp	$0.00010
GPT-4.1-mini	OpenAI	1,515	568 / 5,628	44.0% (33/75) ⋆	±11.2 pp	$0.00040
Gemini-3.1-Flash-Lite	Google	1,818	685 / 3,713	37.3% (28/75)	±10.9 pp	$0.00010
Claude-Haiku-4-5	Anthropic	1,840	946 / 6,484	25.3% (19/75)	±9.8 pp	$0.00080
GPT-5.4	OpenAI	2,324	1,074 / 3,761	25.3% (19/75)	±9.8 pp	$0.01500
Gemini-2.5-Flash	Google	2,589	1,500 / 6,961	25.3% (19/75)	±9.8 pp	$0.00030
Claude-Sonnet-4-6	Anthropic	3,257	2,053 / 6,003	17.3% (13/75) ×	±8.5 pp	$0.00300
Gemini-3-Flash	Google	3,910	2,076 / 9,895	44.0% (33/75) ⋆	±11.2 pp	$0.00030
Gemini-3-Pro	Google	7,146	3,690 / 11,863	38.7% (29/75)	±11.0 pp	$0.00200
Gemini-3.1-Pro	Google	7,595	4,317 / 17,325	33.3% (25/75)	±10.6 pp	$0.00200
Gemini-2.5-Pro	Google	9,094	3,573 / 26,560	40.0% (30/75)	±11.1 pp	$0.00125
GPT-5	OpenAI	13,632	4,197 / 41,941	49.3% (37/75) ⋆	±11.3 pp	$0.01000

- **Conflict detection threshold:** $\tau_{conflict} = 0.35$ (currently 0% effective; lower threshold is the minimal fix).
- **Index strategy:** PARTITIONED (40% worst-case latency reduction at negligible operational cost).
- **Memory top-K policy:** Dynamic (retrieve until relevance < 0.70, hard cap at $K = 15$).
- **LLM for real-time chat:** `GPT-4.1-mini` (44.0% recall, 1,515 ms mean, $0.00040/1k; best latency among top-recall models).
- **LLM for cost-critical deployments:** `Gemini-2.5-Flash-Lite` (40.0% recall, 1,021 ms mean, $0.00010/1k; lowest cost, competitive recall).
- **LLM for offline batch processing:** `GPT-5` (highest recall at 49.3%, latency acceptable for non-real-time workloads).
- **LLM to avoid for memory tasks:** `Claude-Sonnet-4-6` (17.3% recall at $0.00300/1k; worst cost-recall value among all twelve models).

VI. LIMITATIONS AND FUTURE WORK

Scope of recall evaluation: The binary keyword-match recall metric is intentionally simple; a model receives credit only if it mentions the exact expected keyword. A semantic similarity-based evaluation metric would provide a more nuanced recall signal.

Probe set size and statistical power: The expanded benchmark uses 75 probes per model (25 question types × 3 runs), achieving 95% CIs of ±8.5 to ±11.3 pp. Models within 11 pp of each other remain statistically indistinguishable. Expanding to 100+ probes across multiple synthetic user personas would further narrow CIs.

Conflict detection scope: The 20-scenario evaluation set was synthetically constructed. Real-world belief evolution may have substantially different base rates (UPDATE is likely rarer than our 50% representation).

Embedding model and MRL truncation: All experiments used OpenAI text-embedding-3-large truncated to 1536 dimensions via MRL. The five universally failing recall probes likely sit a few hundredths of a point below the 0.60 threshold. This hypothesis should be empirically verified by re-running the benchmark with dimensions=3072. Investigating contradiction-aware fine-tuning of embedding models is a high-priority future direction.

Temporal evaluation: All experiments are single-point-in-time. Longitudinal evaluation of memory quality degradation as user facts change over weeks and months would provide additional deployment-relevant insights.

VII. CONCLUSION

This paper presented a rigorous, empirical investigation of long-term memory architectures for conversational AI agents. Five controlled experiments on a fully open-source system produced the following contributions:

Threshold characterization: Recall@10 spans 88% to 0% across $\tau \in [0.40, 0.90]$. The optimal operating point is $\tau = 0.50$–0.55. Default thresholds of 0.75–0.80 reduce recall to below 20%.

Volume degradation: Memory volume silently degrades retrieval relevance quality by 15.4% (0.838 $\rightarrow$ 0.709) between 50 and 500 stored memories, while retrieved count remains stable masking the degradation from operators.

Index strategy: PARTITIONED per-user Milvus indexing reduces p95 retrieval latency by 28% and worst-case latency by 40% versus FLAT indexing at small mean cost (1.08×), with Cohen's d = 0.21 effect size.

Conflict resolution failure mode: LLM-based conflict resolution achieves 0% accuracy on belief-update and duplicate detection scenarios due to cosine similarity's structural inability to surface semantically opposing facts. 70% of failures occurred despite the LLM reasoning correctly once given the relevant context.

Cross-provider benchmark: Among twelve tested LLMs (n=75 probes per model), recall accuracy ranges from 17.3% (Claude-Sonnet-4-6) to 49.3% (GPT-5). GPT-4.1-mini and Gemini-3-Flash tie at 44.0% recall statistically indistinguishable from GPT-5 while costing 25× and 33× less per token respectively. All twelve models fail identically on five retrieval-threshold-limited probes, confirming that threshold selection dominates model selection as a system design choice.

These findings have direct practical implications for the design and deployment of LLM-based memory systems and

establish a replicable evaluation methodology for the community. The system, datasets, and all experiment scripts are available at the repository accompanying this submission.

References

References

[1] C. Packer et al., "MemGPT: Towards LLMs as Operating Systems," arXiv:2310.08560, 2023.

[2] M. Singh, K. Anand et al., "Mem0: The Memory Layer for AI," GitHub repository, 2024.

[3] LangChain AI, "LangMem: Long-term Memory for LLM Applications," GitHub, 2024.

[4] LangChain AI, "LangGraph: Build Stateful, Multi-Actor Applications with LLMs," GitHub, 2024.

[5] J. S. Park et al., "Generative Agents: Interactive Simulacra of Human Behavior," in *Proc. UIST '23*, 2023.

[6] Zep AI, "Zep: Fast, Scalable Building Blocks for Production LLM Apps," GitHub, 2024.

[7] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Proc. NeurIPS*, 2020.

[8] N. F. Liu et al., "Lost in the Middle: How Language Models Use Long Contexts," *Trans. Assoc. Comput. Linguist.*, vol. 12, pp. 157–173, 2024.

[9] J. Xu, A. Szlam, and J. Weston, "Beyond Goldfish Memory: Long-Term Open-Domain Conversation," in *Proc. ACL*, 2022.

[10] D. Hendrycks et al., "Measuring Massive Multitask Language Understanding," in *Proc. ICLR*, 2021.

[11] M. Chen et al., "Evaluating Large Language Models Trained on Code," arXiv:2107.03374, 2021.

[12] W.-L. Chiang et al., "Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference," in *Proc. ICML*, 2024.

[13] A. Kusupati et al., "Matryoshka Representation Learning," in *Proc. NeurIPS*, 2022.

[14] A. Maharana et al., "Evaluating Very Long-Term Conversational Memory of LLM Agents," in *Proc. ACL*, 2024.