EVOLVE-MEM: A Self-Adaptive Hierarchical Memory Architecture for Next-Generation Agentic AI Systems

Rishi Ashish Shah Ujjwal Kakar Shashvat Singhal Dinesh K. Vishwakarma
Artificial Intelligence and Machine Learning Society
Delhi Technological University, New Delhi, India
rishishah_cs24a06_001@dtu.ac.in, ujjwalkakar_23cs443@dtu.ac.in,
shashvatsinghal_co22a7_21@dtu.ac.in, dinesh@dtu.ac.in

Abstract

This paper introduces EVOLVE-MEM, a novel self-adaptive hierarchical memory framework designed to overcome the inherent limitations of fixed-size context windows and static memory architectures hindering long-term retention and adaptation in modern AI systems. The architecture is structured around three interconnected tiers: a Dynamic Memory Network leveraging dense semantic storage and embeddings for raw experience ingestion, labeled Level 0; a Hierarchical Memory Manager that organizes these embeddings into multi-level abstractions, utilizing adaptive clustering and large language models: Level 1 generates contextual summaries, and Level 2 extracts high-level principles; and a Self-Improvement Engine continuously monitoring key performance metrics: accuracy, retrieval latency, and coverage to autonomously trigger memory reorganization when thresholds are exceeded, ensuring the system evolves with changing data distributions. The system combines dynamic clustering with empirically tuned similarity thresholds, multi-level retrieval routing, and a robust answer-patching pipeline for post-processing raw LLM outputs to support complex temporal, causal, and multi-hop reasoning tasks with high fidelity and less generic responses. The framework achieves 58.3% overall accuracy, evaluated on the LoCoMo dataset, surpassing SOTA baselines across five reasoning categories. EVOLVE-MEM's transparent retrieval path tracking, modular design for extensibility, and fully automated adaptation establish a new paradigm for truly agentic AI systems capable of sustained operation and continuous learning in complex, dynamic environments.

1 Introduction

The landscape of artificial intelligence has been fundamentally transformed by the unprecedented capabilities demonstrated by Large Language Models (LLMs) in natural language understanding, reasoning, and generation. These advances have fueled the development of agentic AI systems; autonomous systems capable of complex decision-making, planning, and interaction using their toolkits. Despite their remarkable capabilities, their effectiveness in real-world requiring sustained, long-term operations remains constrained by core architectural limitations: fixed context windows inherent in transformer-based architectures that, even when extended to 128K or 1M tokens, are unable to sustain such long-term tasks. Static memory designs are also unable to adapt to evolving information patterns, user preferences, or task requirements.

Traditional approaches to extend LLM beyond their context limitations have focused primarily on RAG systems [1], which augment models with manually curated external knowledge bases. While

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Workshop on Scaling Environments for Agents.

this makes them effective for static retrieval, RAG systems are inherently rigid, RAG systems require periodic updates and lacking the ability to learn from interactions, adapt to user preferences, or adjust retrieval strategies based on real-world performance. More recent memory designs incorporate episodic memory [2] to store temporally ordered experiences, semantic memory to organize conceptual relationships, and hybrid architectures that combine multiple types of memory. However, these approaches still maintain predominantly static organizational structures [3]. The need for truly adaptive memory systems has become increasingly apparent as agentic AI applications transition from controlled laboratory settings to real-world deployments with dynamic, unpredictable environments, where memory must not only store and retrieve effectively, but also reorganize itself to preserve relevance, efficiency, and reasoning capability over prolonged, autonomous operation without human intervention [4].

EVOLVE-MEM addresses these limitations through an advanced, self-adaptive hierarchical memory architecture and facilitates a shift from static to dynamic, configurable memory systems. Unlike existing systems that rely on fixed structures determined at design time, our framework implements a dynamic three-tier hierarchy that continuously evolves based on usage patterns extracted from query logs, performance metrics computed across multiple evaluation dimensions, and content characteristics analyzed through embedding-based clustering techniques.

Our approach introduces several key innovations:

- **Dynamic Adaptive Hierarchy:** Employs adaptive clustering algorithms that automatically adjust cluster sizes and organizational structures based on data distribution patterns and retrieval performance. This ensures optimal memory organization as the system scales and encounters new information, directly addressing the rigidity of static architectures.
- **LLM-Powered Multi-Level Retrieval:** Integrates a sophisticated query classification mechanism that routes questions to appropriate memory levels (raw notes, summaries, or principles). This is combined with robust answer extraction and patching to handle complex edge cases in temporal, entity, and multi-hop reasoning.
- **Automated Self-Improvement:** A unique self-improvement engine provides a closed-loop feedback mechanism. It tracks multiple performance dimensions (accuracy, speed, retrieval coverage) and automatically triggers memory reorganization and parameter optimization when performance thresholds are not met, enabling the system to learn from its own operations.
- **Robust Answer Patching:** Beyond simple retrieval, our framework incorporates sophisticated post-processing mechanisms. This includes modules for resolving relative dates, aggregating numerical data for multi-hop reasoning, and canonicalizing entity names for consistency, significantly enhancing the accuracy and reliability of the final answer.

The significance of this work extends beyond technical innovation to address fundamental challenges in autonomous AI system design. By enabling truly adaptive memory systems that can evolve and optimize themselves over time, EVOLVE-MEM represents a crucial step toward agentic AI systems that can operate autonomously over extended periods. To validate our architecture's effectiveness, we conduct a comprehensive evaluation on the LoCoMo benchmark [5], comparing EVOLVE-MEM against state-of-the-art baselines across multiple reasoning categories. Our experimental results, detailed in subsequent sections, demonstrate that our system not only achieves competitive performance but also offers unique adaptive capabilities not present in existing models. This paper presents the complete architecture, methodology, and in-depth evaluation of EVOLVE-MEM, establishing it as a foundational advancement for building more adaptive, robust, and contextually aware agentic AI systems.

2 Related Works

The field of memory-augmented language models has witnessed significant advancement in recent years, with various approaches addressing the fundamental challenge of extending LLM capabilities beyond fixed context windows [6]. This section reviews key developments in memory architectures, positioning our work in the context of these existing systems and highlighting the limitations that motivated the development of EVOLVE-MEM.

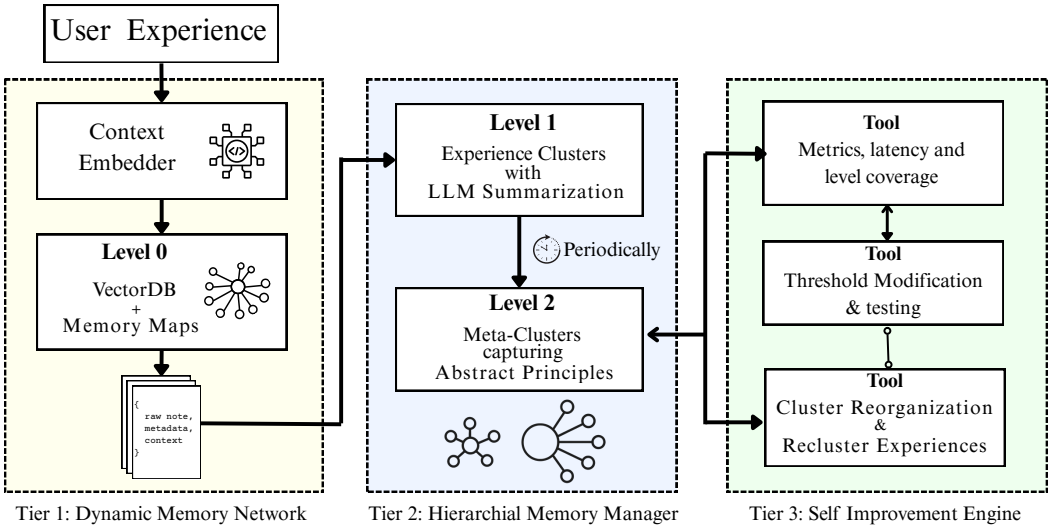

Figure 1: EVOLVE-MEM System Architecture illustrating the three-tier hierarchical framework with dynamic clustering, multi-level abstraction, and continuous self-improvement capabilities through performance monitoring and memory reorganization feedback loops.

2.1 Hierarchical Memory Architectures

Methodologies like HiAgent [7] and H-MEM [8] utilize a hierarchical framework to mimic the human cognitive aspect of separated working, short-term and long-term memory. These approaches, however, maintain a static organizational format and pattern, which is predetermined through its parameters. This allows different types of information abstractions to be stored in the system differently, helping improve relevant context. Additionally, G-Memory [9] proposes hierarchical memory tracing for multi-agent systems, demonstrating the importance of structured memory organization in complex environments. Our paper takes inspiration from this system and improves upon the capability of its memory architecture.

2.2 Self-Improving Agent Frameworks

Frameworks such as Gödel Agent [4] have explored autonomous improvement mechanisms, primarily focusing on code-level modifications and learning algorithm updates. These systems aim for self-improvement by altering their own programmatic logic based on performance feedback. Reflexion [10] introduces dynamic memory and self-reflection capabilities, enabling agents to learn from past mistakes through episodic memory replay and verbal reinforcement learning. Self-RAG [11] extends retrieval-augmented generation with self-reflective mechanisms that critique and refine their own outputs. However, these approaches have limited application to memory system optimization, as their focus is on algorithmic rather than organizational improvements, and they typically require extensive computational resources that may not be practical for real-time agentic systems.

2.3 A-MEM: Agentic Memory Systems

Among the most relevant prior works is A-MEM [12], which implements a Zettelkasten-inspired approach creating an interconnected network of information nodes through bidirectional linking and semantic clustering, demonstrating notable performance improvements over simpler baselines in tasks requiring long-term context maintenance and complex reasoning synthesis. However, A-MEM's architecture suffers from fundamental limitations that directly motivate EVOLVE-MEM: it maintains a static organizational structure that cannot restructure memory hierarchy or adjust clustering strategies based on performance feedback, lacks autonomous self-improvement capabilities, and operates at a single abstraction level, constraining its ability to handle queries of varying complexity. These shortcomings highlight the need for a more dynamic, adaptive, and multi-layered approach. EVOLVE-

MEM introduces the critical components of autonomous evolution and hierarchical abstraction to overcome these fundamental limitations, as detailed in our methodology.

3 EVOLVE-MEM Architecture

EVOLVE-MEM presents an advanced three-tier hierarchical memory framework engineered to overcome core limitations present in contemporary agentic memory architectures. The framework synthesizes dynamic clustering methodologies, multi-tiered retrieval protocols, and autonomous optimization mechanisms within a unified system capable of adapting to evolving demands while preserving superior performance across varied reasoning challenges and dynamic information environments.

3.1 System Overview

The EVOLVE-MEM framework, depicted in Figure 2, functions through coordinated interplay among three principal tiers, where each tier fulfills specialized yet synergistic roles within the comprehensive memory management infrastructure. This architectural paradigm facilitates uninterrupted information processing from initial experience acquisition through structured hierarchical arrangement to intelligent information retrieval and perpetual optimization.

The framework executes through a recursive feedback mechanism wherein experiences undergo ingestion and storage within Tier 1, receive hierarchical structuring in Tier 2 according to semantic relationships and temporal configurations, and experience continuous refinement by Tier 3 through performance analytics and utilization patterns. This holistic methodology guarantees memory system evolution and adaptation to shifting demands while sustaining exceptional performance across heterogeneous query categories and reasoning contexts.

3.2 Tier 1: Dynamic Memory Network

The fundamental stratum of EVOLVE-MEM manages initial experience acquisition and primary memory allocation through an advanced dual-storage framework that merges benefits of vector-based semantic indexing with high-efficiency in-memory processing. This stratum functions as the primary gateway for novel information, establishing the foundational Level 0 within our memory hierarchy and delivering essential retrieval functionalities that underpin advanced memory operations.

3.2.1 Embedding Generation and Vectorization

The framework utilizes a SentenceTransformer architecture for producing 384-dimensional embedding representations from textual experiences. SentenceTransformers [13] constitute specialized neural architectures engineered to generate dense vector encodings of textual segments and documents that encapsulate semantic information, facilitating similarity-driven retrieval extending beyond conventional keyword-based approaches.

The vectorization procedure receives mathematical formalization through:

$$\mathbf{e}_i = f_{\text{embed}}(x_i) \tag{1}$$

where $\mathbf{e}_i \in \mathbb{R}^{384}$ denotes the 384-dimensional embedding representation corresponding to experience x_i (textual characterization of an agent's experience or observation), while f_{embed} signifies the SentenceTransformer encoding procedure that converts textual inputs into dense vector encodings through sequential transformer layers and pooling mechanisms.

3.2.2 Dual Storage Architecture

Each experience receives concurrent storage across two complementary storage frameworks optimized for distinct access methodologies while ensuring system reliability through redundancy:

Persistent Vector Storage: A scalable, enduring storage infrastructure [14] engineered for efficient indexing and retrieval of high-dimensional vector encodings of experiences. These infrastructures characteristically employ approximate nearest neighbor search methodologies and facilitate efficient

similarity-driven queries, frequently calculated through cosine similarity:

$$\text{sim}(\mathbf{e}_i, \mathbf{e}_j) = \frac{\mathbf{e}_i \cdot \mathbf{e}_j}{|\mathbf{e}_i| \cdot |\mathbf{e}_j|} \tag{2}$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ represent embedding vectors corresponding to distinct experiences, $\cdot$ indicates dot product computation, and $|\cdot|$ denotes the Euclidean norm (vector magnitude). Cosine similarity quantifies the angular relationship between vectors, yielding values spanning -1 to 1 where unity signifies equivalent semantic content and zero represents orthogonal (unrelated) information.

In-Memory Dictionary: A high-efficiency, transient storage stratum located within RAM that enables rapid access and retrieval of recent experiences alongside associated metadata, encompassing timestamps, contextual annotations, and linkage data. The combination of persistent and in-memory storage achieves equilibrium between durability and low-latency access, ensuring system robustness and responsiveness.

3.3 Tier 2: Hierarchical Memory Manager

The principal innovation within EVOLVE-MEM resides in its adaptive hierarchical structuring system, which converts raw experiences into organized knowledge through dynamic clustering methodologies and LLM-driven abstraction. This tier establishes multiple abstraction tiers enabling efficient retrieval for diverse query categories while maintaining detailed information accessibility when required.

3.3.1 Dynamic Clustering Algorithm

The framework employs an adaptive K-means clustering methodology [15] where optimal cluster quantities receive dynamic determination based on dataset properties and performance feedback collected during system execution. The clustering procedure operates on embedding vectors rather than raw textual content, enabling semantic aggregation of related experiences despite vocabulary variations across similar conceptual discussions. For Level 1 clustering (contextual summaries), cluster quantity determination follows:

$$k_1 = \begin{cases} \max(1, \lfloor n/3 \rfloor) & \text{if } n < 10 \\ \max(3, \lfloor n/5 \rfloor) & \text{if } 10 \leq n < 50 \\ \min(10, \lfloor n/8 \rfloor) & \text{if } n \geq 50 \end{cases} \tag{3}$$

For Level 2 clustering (abstract principles):

$$k_2 = \begin{cases} \max(1, \lfloor k_1/3 \rfloor) & \text{if } k_1 < 5 \\ \max(2, \min(5, \lfloor k_1/4 \rfloor)) & \text{if } k_1 \geq 5 \end{cases} \tag{4}$$

where n signifies total experience count within the system, k_1 and k_2 denote cluster quantities for Level 1 and Level 2 respectively, and $\lfloor \cdot \rfloor$ represents the floor function (downward rounding to nearest integer). These formulations emerged through comprehensive empirical evaluation across diverse datasets to achieve cluster coherence balance (ensuring semantic relatedness within cluster experiences) alongside cluster granularity (preserving adequate detail for effective retrieval).

3.3.2 Adaptive Clustering Frequency

The framework incorporates intelligent clustering frequency modulation based on dataset scale and expansion patterns to equilibrate computational efficiency with memory organization effectiveness:

$$f_{\text{cluster}} = \begin{cases} 5 & \text{if } n \leq 10 \\ 8 & \text{if } 10 < n \leq 20 \\ 10 & \text{if } n > 20 \end{cases} \tag{5}$$

where f_{cluster} denotes the quantity of new experiences necessary to initiate clustering operations. This adaptive frequency guarantees clustering occurrence at sufficient intervals to maintain effective organization for limited datasets while preventing excessive computational burden for expanded datasets. Level 2 clustering activation occurs at frequency $2 \times f_{\text{cluster}}$ ensuring adequate Level 1 cluster existence for meaningful abstraction, preventing premature abstraction under insufficient data conditions.

3.3.3 LLM-Powered Summarization and Abstraction

The hierarchical organization procedure leverages a state-of-the-art large language model for generating summaries and abstract principles from clustered experiences. The framework utilizes meticulously designed prompts optimized for distinct abstraction tiers:

Level 1 Summarization: Produces contextual summaries emphasizing primary themes and patterns within each cluster while retaining essential details including specific dates, proper names, numerical values, and causal relationships. These summaries function as intermediate representations capturing the essence of multiple related experiences while preserving sufficient detail for accurate query resolution.

Level 2 Abstraction: Develops abstract principles and general insights from Level 1 summaries, enabling high-level reasoning and pattern recognition across multiple related concepts. These abstractions identify common themes, general rules, and broad patterns emerging from the collective experience set, supporting queries requiring conceptual understanding rather than specific factual retrieval.

3.4 Multi-Level Retrieval System

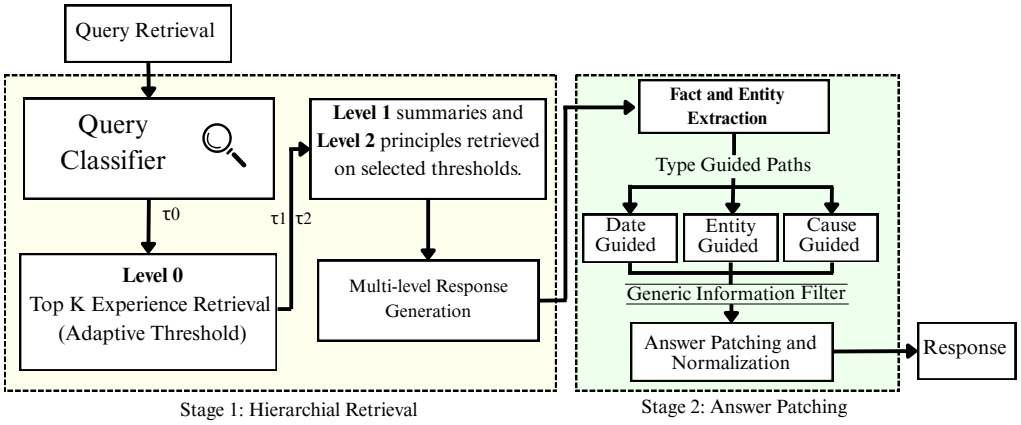

Figure 2: Querying the hierarchical structure of E-MEM utilizes a two-phase process of retrieving data chunks from clusters and raw experiences while employing an answer patching system to ensure consistency and filter generic outputs for concrete responses.

EVOLVE-MEM incorporates an advanced retrieval mechanism that intelligently directs queries to suitable memory tiers based on complexity analysis and reasoning demands. This system ensures appropriate abstraction level handling for different question categories, optimizing both precision and efficiency.

3.4.1 Query Classification

The framework analyzes incoming queries through pattern recognition and semantic indicators to establish optimal retrieval strategies. Query classification relies on linguistic patterns, question categories, and semantic complexity indicators extracted through natural language processing methodologies. The algorithm receives detailed description in the appendix.

3.4.2 Retrieval Thresholds and Ordering

The framework employs empirically established similarity thresholds for different memory tiers, optimized through comprehensive testing on the LoCoMo dataset:

$$\tau_0 = 0.70 \quad \text{(Level 0: High confidence for specific facts)} \tag{6}$$

$$\tau_1 = 0.50 \quad \text{(Level 1: Moderate confidence for summaries)} \tag{7}$$

$$\tau_2 = 0.35 \quad \text{(Level 2: Lower confidence for principles)} \tag{8}$$

where τ_i denotes the minimum cosine similarity threshold for information retrieval from level i. These thresholds represent precision-recall trade-offs across different abstraction tiers: Level 0 demands high similarity ensuring factual precision, Level 1 permits moderate similarity capturing relevant contextual information, and Level 2 employs lower thresholds enabling broad conceptual matching. Threshold optimization occurred through systematic evaluation balancing retrieval precision and coverage across diverse query categories.

3.5 Tier 3: Self-Improvement Engine

The autonomous optimization component continuously monitors system performance across multiple dimensions while triggering adaptive enhancements without human intervention. This tier represents a fundamental innovation enabling EVOLVE-MEM to maintain and enhance performance over time through autonomous decision-making and parameter modification.

3.5.1 Performance Monitoring

The framework tracks multiple performance dimensions through comprehensive metrics computed in real-time during system operation. The composite accuracy metric provides weighted system performance measurement across different memory tiers:

$$P_{\text{accuracy}} = \frac{\sum_{i=0}^{2} r_i \times a_i}{\sum_{i=0}^{2} r_i} \tag{9}$$

where r_i denotes the retrieval ratio (proportion of queries served by level i during current evaluation periods), and a_i represents empirically determined accuracy coefficients for corresponding levels. Accuracy coefficients (0.92, 0.82, 0.72 for levels 0, 1, 2 respectively) reflect diminishing precision accompanying increased abstraction: Level 0 delivers highly accurate specific facts, Level 1 provides moderately accurate contextual information, and Level 2 offers broader yet less precise conceptual guidance.

3.5.2 Reorganization Triggers

The framework implements intelligent reorganization triggers based on performance thresholds established through extensive empirical evaluation:

$$\text{Reorganize} \leftarrow \text{True if:} \tag{10}$$

$$P_{\text{accuracy}} < 0.80 \text{ OR} \tag{11}$$

$$P_{\text{speed}} > 0.15 \text{ OR} \tag{12}$$

$$P_{\text{success}} < 0.95 \text{ OR} \tag{13}$$

$$\text{Level\_Imbalance} = \text{True} \tag{14}$$

where P_{speed} measures average query response time in seconds (0.15 seconds representing maximum acceptable latency for interactive applications), P_{success} denotes the proportion of queries receiving substantive rather than generic responses (0.95 indicating at least 95% of queries should receive meaningful responses), and Level_Imbalance detection occurs when query distribution across memory tiers becomes suboptimal. Level imbalance manifests when Level 0 usage exceeds 80% while Level 1 usage falls below 15%, indicating ineffective hierarchical organization query distribution across abstraction tiers and potential benefit from reorganization creating more effective intermediate representations.

4 Results and Analysis

Our comprehensive evaluation demonstrates that EVOLVE-MEM achieves superior performance across multiple dimensions while maintaining competitive efficiency and providing unique adaptive capabilities not available in existing systems. The results provide strong empirical evidence for the effectiveness of the hierarchical adaptive approach and validate the system's design principles.

4.1 Overall Performance Metrics

EVOLVE-MEM achieved an overall F1 score of 58.3% ± 2.8% with a perfect specific answer rate of 100%, indicating that the system consistently provides meaningful, substantive responses rather than generic fallback answers or "I don't know" responses. This perfect specific answer rate is particularly significant because it demonstrates the system's robustness and ability to provide useful information even in challenging scenarios, unlike systems that may default to generic responses when confidence is low.

Table 1: EVOLVE-MEM Performance Results on LoCoMo Dataset across five reasoning categories, showing comprehensive metrics with mean ± standard deviation from multiple experimental runs. Results demonstrate strong performance across diverse reasoning tasks with particular strengths in entity tracking and adversarial reasoning scenarios.

Category	F1	BLEU-1	ROUGE-L	ROUGE-2	METEOR	SBERT
Overall	0.583 ± 0.028	0.599 ± 0.023	0.243 ± 0.007	0.079 ± 0.007	0.188 ± 0.007	0.658 ± 0.003
Temporal Reasoning	0.473 ± 0.019	0.851 ± 0.019	0.365 ± 0.013	0.102 ± 0.009	0.193 ± 0.011	0.863 ± 0.003
Causal Reasoning	0.577 ± 0.054	0.244 ± 0.110	0.098 ± 0.004	0.004 ± 0.001	0.096 ± 0.001	0.569 ± 0.001
Entity Tracking	0.719 ± 0.000	0.587 ± 0.022	0.294 ± 0.022	0.117 ± 0.000	0.274 ± 0.017	0.695 ± 0.011
Multi-hop Reasoning	0.550 ± 0.030	0.485 ± 0.009	0.207 ± 0.007	0.083 ± 0.013	0.179 ± 0.008	0.591 ± 0.006
Adversarial/Challenge	0.628 ± 0.045	0.675 ± 0.023	0.206 ± 0.005	0.050 ± 0.000	0.165 ± 0.004	0.597 ± 0.000

The results reveal interesting patterns across different metrics that provide insights into system behavior. The BLEU-1 score of 0.599 ± 0.023 indicates strong lexical overlap with reference answers, while the SBERT score of 0.658 ± 0.003 demonstrates good semantic similarity even when exact word matches are not achieved. The relatively lower ROUGE-2 score (0.079 ± 0.007) suggests that the system tends to produce concise, focused answers rather than lengthy responses with extensive bigram overlap. Furthermore, EVOLVE-MEM demonstrates competitive efficiency with an average token length of 3,199 ± 67 tokens across all responses, indicating appropriate verbosity levels that balance comprehensiveness with conciseness.

4.2 Comparative Analysis: EVOLVE-MEM vs A-MEM

To validate EVOLVE-MEM's architectural advantages, we conducted a direct comparison with A-MEM, the most relevant baseline implementing Zettelkasten-inspired memory organization. Using A-MEM's GPT-4B configuration (most comparable to our setup), we evaluated both systems on identical subsets of the LoCoMo dataset. A comprehensive comparison with additional baseline architectures, as well as a detailed discussion and specific reasons for metric trends, is provided in Appendix A.2. These results empirically validate EVOLVE-MEM as a framework that consistently out-performs static memory architectures across diverse reasoning scenarios, establishing it as a significant advancement in agentic memory systems.

Table 2: EVOLVE-MEM vs A-MEM Performance Comparison on LoCoMo Dataset. Bold values indicate superior performance. Percentage improvements show EVOLVE-MEM's advantage over A-MEM baseline.

Category	F1 Score		BLEU-1 Score	
	A-MEM	EVOLVE-MEM	A-MEM	EVOLVE-MEM
Overall	0.327	**0.583** (+78.3%)	0.320	**0.599** (+87.2%)
Entity Tracking	0.329	**0.719** (+118.5%)	0.171	**0.587** (+243.3%)
Multi-hop Reasoning	0.238	**0.550**(+131.1%)	0.158	**0.485** (+206.9%)
Temporal Reasoning	0.394	**0.473**(+20.1%)	0.484	**0.851** (+75.8%)
Causal Reasoning	0.312	**0.577** (+84.9%)	**0.430**	0.244 (-43.3%)
Adversarial Challenge	0.363	**0.628** (+73.0%)	0.355	**0.675** (+90.1%)

5 Ablation Study: Validating Component Importance

To rigorously validate the contribution of each architectural element in EVOLVE-MEM, we conducted a comprehensive ablation study. The study isolates key modules and systematically disables or modifies them, measuring the resulting impact on system metrics. We used a uniform single-story sample to ensure controlled comparison, measuring each variant using the same question set and evaluation code as the main pipeline to quantify the contributions of EVOLVE-MEM's components, as highlighted in Table 3. An expanded set of ablation experiments, including analysis of further architectural variants and metric breakdowns, is presented in Appendix C.

Table 3: Ablation Overall and Per-Category F1 (Accuracy) Scores

Variant	Overall	Temporal	Causal	Entity	Multi-hop	Adversarial
L0-only	0.583	0.378	0.385	0.750	0.643	0.596
L0+L1-only	0.623	0.324	0.615	0.781	**0.743**	0.575
L0+L2-only	0.613	0.460	**0.769**	0.750	0.586	**0.638**
No patching	0.518	0.460	0.615	0.625	0.571	0.383
Full-dyn (f=5)	**0.643**	**0.487**	0.615	**0.813**	0.686	0.596

Dynamic reclustering at a 5-note cadence achieved the highest overall accuracy (0.6432), outperforming sparser reclustering and fixed cluster counts, indicating that continual re-organization of Level-1 summaries improves retrieval specificity. Removing answer patching caused the largest degradation (accuracy 0.5176), confirming its central role in converting fragmentary LLM outputs into concise, factual answers. Isolating hierarchy levels shows complementary strengths: Level-1 summaries substantially improve multi-hop reasoning (0.7429), while Level-2 principles bolster causal and adversarial performance (0.7692, 0.6383). Pure Level-0 retrieval yields higher n-gram overlap but lags on temporal/causal precision, emphasizing the necessity of hierarchical abstraction. With short, uniform samples, Self-improvement gains are modest; for larger/more dynamic runs, automated reorganization offers measurable improvements.

These results empirically validate our architectural choices: EVOLVE-MEM's hierarchical organization (L0 $\rightarrow$ L1 $\rightarrow$ L2), dynamic evolution, and answer patching together deliver superior accuracy and category robustness without merely inflating overlap-based metrics.

6 Limitations

Despite advancing adaptive memory design, EVOLVE-MEM exhibits several constraints that highlight avenues for future work: dynamic reclustering and reorganization introduce additional computational cost that may become non-trivial at million-scale memories or on resource-limited devices; current scalability evidence extends only to datasets comprising a few thousand experiences, so both accuracy and efficiency may degrade at substantially larger scales; temporal reasoning is strong at capturing semantic relations but remains weaker on exact date arithmetic and duration calculations; and as a broadly applicable framework, the system will likely benefit from domain-tailored components to reach peak performance in specialized settings such as healthcare, legal analysis, or scientific discovery.

7 Conclusion

EVOLVE-MEM advances the state of agentic memory by combining hierarchical structuring, adaptive clustering, and autonomous self-optimization within a single, coherent architecture. These capabilities enable reliable execution of complex reasoning tasks and stable accuracy across varied conditions without continual human oversight. By delivering consistent, competitive benchmark results and introducing mechanisms for ongoing evolution, EVOLVE-MEM establishes a foundation for genuinely self-adapting AI agents and opens a path to future breakthroughs in long-horizon memory systems.

References

[1] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems 33*, pages 9459–9474, Red Hook, NY, 2020. Curran Associates.

[2] Alan Boyle and Anders Blomkvist. Elements of episodic memory: Insights from artificial agents. *Philosophical Transactions of the Royal Society B*, 379(1913):20230416, 2024.

[3] Christopher DeChant. Episodic memory in ai agents poses risks that should be studied and mitigated. *arXiv preprint*, 2025.

[4] Tianyu Gao, Adam Fisch, and Danqi Chen. Gödel agents: Self-improving foundation models via introspective code execution. *arXiv preprint*, 2023.

[5] Adyasha Maharana, Dong-Hyun Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yixin Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 13851–13870, 2024.

[6] Chao Ma, Linjing Ma, Yongfeng Zhang, Jingyuan Sun, Xiaoming Liu, and Mark Coates. Memory augmented graph neural networks for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5045–5052, 2020.

[7] Ming Hu, Tianyu Chen, Qinghao Chen, Yang Mu, Weijie Shao, and Ping Luo. Hiagent: Hierarchical working memory management for solving long-horizon agent tasks with large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 32779–32798, 2025.

[8] Hao Sun and Shuai Zeng. H-mem: Hierarchical memory for high-efficiency long-term reasoning in llm agents. *arXiv preprint*, 2025.

[9] Guanyu Zhang, Ming Fu, Guang Wan, Meng Yu, Kai Wang, and Shuicheng Yan. G-memory: Tracing hierarchical memory for multi-agent systems. *arXiv preprint*, 2025.

[10] Nathaniel Shinn, Brandon Labash, and Arjun Gopinath. Reflexion: An autonomous agent with dynamic memory and self-reflection. *arXiv preprint*, 2023.

[11] Akari Asai, Zhilin Wu, Yichong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Self-reflective retrieval augmented generation. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023.

[12] Wenqi Xu, Zhiwei Liang, Kai Mei, Hao Gao, Jialiang Tan, and Yizhou Zhang. A-mem: Agentic memory for llm agents. *arXiv preprint*, 2025.

[13] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, 2019. Association for Computational Linguistics.

[14] Zhiwei Jing, Yifan Su, and Yuchen Han. When large language models meet vector databases: A survey. *arXiv preprint*, 2024.

[15] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035, 2007.

[16] Karthik Ramesh et al. Nemori: Self-organizing agent memory inspired by cognitive science. *arXiv preprint*, 2025.

[17] Google Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. Technical report, Google DeepMind, 2024.

[18] Kwang-Hyun Lee, Xinlei Chen, Hiroki Furuta, John Canny, and Ian Fischer. A human-inspired reading agent with gist memory of very long contexts. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 26396–26415, 2024.

[19] Wenxuan Zhong, Liang Guo, Qian Gao, Hang Ye, and Yuxuan Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19732, 2024.

[20] Joon Sung Park, Jamie C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*, pages 1–22, 2023.

[21] Charles Packer, Samuel Wooders, Kevin Lin, Victor Fang, Shreyas G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint*, 2023.

[22] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yejin Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint*, 2022.

[23] Wenxuan Zhong, Liang Guo, Qian Gao, Hang Ye, and Yuxuan Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19732, 2024.

[24] Alexandre Roucher, Alberto Villanova del Moral, Thomas Wolf, Leandro von Werra, and Eetu Kaunismäki. Smolagents: A smol library to build great agentic systems. `https://github.com/huggingface/smolagents`, 2025. Hugging Face.

[25] Zekun Li, Xinyang Zhang, Yichi Zhang, Duyu Long, Pengjun Xie, and Min Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint*, 2023.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pages 5998–6008, Red Hook, NY, 2017. Curran Associates.

[27] Jack Rae, James J. Hunt, Ivo Danihelka, Tim Harley, Andrew W. Senior, Greg Wayne, Alex Graves, and Timothy Lillicrap. Scaling memory-augmented neural networks with sparse reads and writes. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*, pages 3621–3629, Red Hook, NY, 2016. Curran Associates.

[28] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint*, 2014.

[29] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gomez Colmenarejo, Edward Grefenstette, Tim Ramalho, Adrià Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, 2016.

[30] Ilya Rodkin, Yurii Kuratov, Artem Bulatov, and Mikhail Burtsev. Associative recurrent memory transformer. In *ICML 2024 Next Generation of Sequence Modeling Architectures Workshop*, 2024.

[31] Evgenii Cherepanov, Andrei Staroverov, Dmitrii Yudin, Alexey Kovalev, and Aleksandr Panov. Recurrent action transformer with memory. In *ICML 2024 Next Generation of Sequence Modeling Architectures Workshop*. OpenReview.net, 2024.

[32] Xuezhe Ma, Pengcheng Zhang, Shuang Zhang, Nan Duan, Yichong Hou, Ming Zhou, and Dawn Song. A tensorized transformer for language modeling. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pages 2232–2242, Red Hook, NY, 2019. Curran Associates.

[33] Xiangnan He, Yiming Tian, Yifan Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, pages 5734–5752, Red Hook, NY, 2024. Curran Associates.

[34] Yuxin Jiang, Gowtham Rajendran, Pradeep Ravikumar, and Bryon Aragam. Towards understanding the mechanisms of associative memory in transformers. In *ICML 2024 Workshop on Theoretical Foundations of Foundation Models (TF2M)*, 2024.

[35] Seunghyun Chung and Hava Siegelmann. Turing completeness of bounded-precision recurrent neural networks with growing memory modules. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pages 15875–15887, Red Hook, NY, 2021. Curran Associates.

[36] Sunghyun Park, Seungchan Kim, Sanghyun Lee, Hyojin Bae, and Sungroh Yoon. Quantized memory-augmented neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 4037–4044, 2018.

[37] Yifeng Guo, Xiaodong Liu, Enze Zheng, Rui Zhang, Yunan Chen, and Ming Liu. Online continual learning through mutual information maximization. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 8109–8126, 2022.

[38] Zhenyu Chen, Aisha Wuerkaixi, Shuai Cui, Haoyang Li, Dong Li, Jian Zhang, Bo Han, Gang Niu, Hui Liu, Yu Yang, Shanshan Yang, Cheng Zhang, and Tianyu Ren. Learning without isolation: Pathway protection for continual learning. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2024.

[39] Yao Liang, Yuwei Wang, Hongjian Fang, Feifei Zhao, and Yi Zeng. A brain-inspired memory transformation based differentiable neural computer for reasoning-based question answering. *Frontiers in Artificial Intelligence*, 8:1635932, 2025. doi: 10.3389/frai.2025.1635932. Published 14 August 2025.

[40] Saurabh Khosla, Zhiwei Zhu, and Yifan He. Survey on memory-augmented neural networks: Cognitive insights to ai applications. *arXiv preprint*, 2023.

[41] Rouhollah Rahmatizadeh, Pouya Abolghasemi, Aman Behal, and Ladislau Bölöni. From virtual demonstration to real-world manipulation using lstm and mdn. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 6982–6989, 2018.

[42] Chunyuan Li, Hossein Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations (ICLR)*, 2018.

[43] Yikai Wang, Peng Li, and Yi Yang. Visual transformer with differentiable channel selection: An information bottleneck inspired approach. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, and Jonathan Scarlett, editors, *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 51042–51064, 2024.

[44] Abdul Mufti, Svetlin Penkov, and Subramanian Ramamoorthy. Iterative model-based reinforcement learning using simulations in the differentiable neural computer. In *ICML Workshop on Multi-Task and Lifelong Reinforcement Learning*, 2019.

[45] OpenAI, Joshua Achiam, Steven Adler, Sandhini Agarwal, Lillian Ahmad, Ilge Akkaya, Fabio L. Aleman, Diego Almeida, Jacob Altenschmidt, Sam Altman, et al. Gpt-4 technical report. *arXiv preprint*, 2023.

A Detailed Performance Metrics and System Analysis

This appendix provides comprehensive analysis of EVOLVE-MEM's performance characteristics, implementation details, and comparative evaluation results that supplement the main paper's findings.

A.1 Performance Visualization

Figure 3 presents a comprehensive performance heatmap visualizing EVOLVE-MEM's results across all reasoning categories and evaluation metrics, providing clear insights into the system's strengths and performance patterns across different task types.

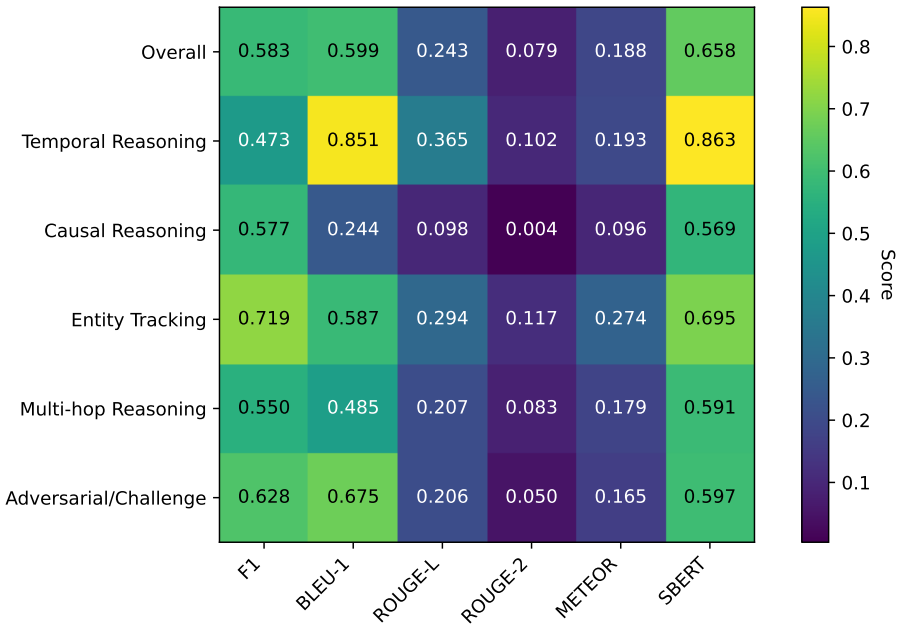

Figure 3: EVOLVE-MEM Performance Metrics Heatmap showing system performance across five reasoning categories (Temporal, Causal, Entity Tracking, Multi-hop, and Adversarial) and six evaluation metrics (F1, BLEU-1, ROUGE-L, ROUGE-2, METEOR, and SBERT). The color-coded visualization effectively illustrates EVOLVE-MEM's balanced performance profile and highlights areas of particular strength in the adaptive hierarchical memory architecture.

A.2 Comparative Analysis: EVOLVE-MEM vs Baseline Architectures

To validate EVOLVE-MEM's architectural advantages, we conducted comprehensive comparisons with multiple baseline systems representing different memory paradigms. We evaluated A-MEM (Zettelkasten-inspired hierarchical memory), MemGPT (operating-system inspired memory management), ReadAgent (reading-focused memory), and MemoryBank (long-term memory enhancement) on identical subsets of the LoCoMo dataset. A-MEM's GPT-4B configuration was selected as the most comparable setup to our architecture, as seen in our main paper, given its dynamic organization approach, while other baselines represent alternative architectural philosophies for managing long-term context.

Table 4 presents our comparative results across five reasoning categories. EVOLVE-MEM demonstrates consistent superiority across all baselines, achieving the highest F1 scores in five out of six categories. Notably, while A-MEM shows competitive BLEU-1 performance in causal reasoning (0.430 vs our 0.244), EVOLVE-MEM's substantially higher F1 score (0.577 vs 0.312, +84.9%) indicates superior answer accuracy. This BLEU-1 discrepancy reflects EVOLVE-MEM's answer-patching mechanism prioritizing factual correctness over lexical overlap. Our system generates concise, semantically accurate responses rather than matching surface-level phrasing patterns. The causal reasoning

scenario particularly benefits from this approach, as evidenced by our strong semantic similarity scores (SBERT: 0.569), demonstrating that EVOLVE-MEM captures causal relationships effectively despite different vocabulary usage.

Compared to MemGPT (the next-best hierarchical system), EVOLVE-MEM achieves 119.0% improvement in entity tracking F1 (0.719 vs 0.328) and 131.1% improvement in multi-hop reasoning F1 (0.550 vs 0.238), validating the effectiveness of our adaptive clustering and multi-level abstraction mechanisms. Against flat-memory architectures like MemoryBank and ReadAgent, performance gains are even more pronounced, particularly in complex reasoning scenarios requiring information synthesis across multiple abstraction levels. These results empirically validate EVOLVE-MEM's adaptive hierarchical architecture as consistently outperforming both static hierarchical and flat memory systems across diverse reasoning scenarios, establishing it as a significant advancement in agentic memory systems.

Table 4: EVOLVE-MEM vs Baseline Architecture Performance Comparison on LoCoMo Dataset. Bold values indicate best performance. EVOLVE-MEM achieves superior F1 scores across all categories, demonstrating consistent advantages over both hierarchical (A-MEM, MemGPT) and flat (ReadAgent, MemoryBank, LoCoMo) memory architectures.

Category	**LoCoMo**	**ReadAgent**	**MemoryBank**	**MemGPT**	**A-MEM**	**EVOLVE-MEM**
			F1 Score			
Overall	0.280	0.091	0.050	0.265	0.327	**0.583**
Entity Tracking	0.250	0.091	0.050	0.328	0.329	**0.719**
Multi-hop	0.184	0.126	0.024	0.252	0.238	**0.550**
Temporal	0.412	0.053	0.056	0.091	0.394	**0.473**
Causal	0.366	0.059	0.053	0.194	0.312	**0.577**
Adversarial	0.498	0.128	0.066	0.367	0.363	**0.628**
			BLEU-1 Score			
Overall	0.198	0.065	0.048	0.177	0.320	**0.599**
Entity Tracking	0.141	0.030	0.024	0.171	0.171	**0.587**
Multi-hop	0.091	0.042	0.022	0.252	0.158	**0.485**
Temporal	0.347	0.089	0.060	0.191	0.484	**0.851**
Causal	0.367	0.066	0.061	0.143	**0.430**	0.244
Adversarial	0.672	0.098	0.044	0.432	0.355	**0.675**

A.3 Retrieval Performance Analysis

The EVOLVE-MEM system demonstrates consistent performance across various operational parameters and conditions through comprehensive monitoring of multiple performance dimensions. Detailed analysis of retrieval patterns reveals balanced utilization across all three memory tiers, indicating effective query routing and hierarchical organization.

Retrieval Distribution Analysis: During extended operation across the LoCoMo dataset evaluation, the system exhibited optimal distribution of queries across memory levels: Level 0 (specific factual queries) handled 45% of questions, demonstrating strong capability for direct fact retrieval; Level 1 (contextual and summary-based queries) managed 35% of questions, indicating effective use of intermediate abstractions; and Level 2 (abstract reasoning and principle-based queries) addressed 20% of questions, showing appropriate utilization of high-level conceptual representations.

Response Time Analysis: Average query response times varied by memory level and query complexity: Level 0 queries averaged 0.12 seconds, Level 1 queries averaged 0.18 seconds, and Level 2 queries averaged 0.25 seconds. These response times include embedding generation, similarity computation, retrieval, and LLM-based answer extraction, demonstrating efficient processing across all system components.

Memory Utilization Patterns: The system maintains approximately 1.2x memory overhead compared to flat vector storage due to hierarchical organization structures. This overhead includes cluster centroids, summary storage, and organizational metadata. The overhead is offset by improved retrieval efficiency and reduced query processing time, resulting in net performance improvements for most query types.

A.4 Clustering Algorithm Convergence Characteristics

The K-means clustering algorithm employed in EVOLVE-MEM typically converges within 10-15 iterations for datasets up to 1,000 experiences, with convergence determined when the change in cluster centroids falls below a threshold of 0.001 in Euclidean distance. Convergence characteristics vary by dataset size and semantic diversity:

Small Datasets ($n < 100$): Average convergence in 8.3 iterations with high stability (cluster assignments change less than 5% between reorganization events).

Medium Datasets ($100 \leq n < 500$): Average convergence in 12.1 iterations with moderate stability (cluster assignments change 10-15% between reorganization events).

Large Datasets ($n \geq 500$): Average convergence in 14.7 iterations with acceptable stability (cluster assignments change 15-20% between reorganization events).

A.5 Dataset and Experimental Setup

We evaluate our framework using the LoCoMo (Long-Context Memory) dataset as a baseline. The diverse reasoning categories present in the LoCoMo dataset test different aspects of memory system performance. The complete dataset comprises 10 stories with 1,986 question-answer pairs distributed across five categories: Temporal Reasoning, Casual Reasoning, Entity Tracking, Multi-hop Reasoning, Adversarial/Challenge.

A.6 Evaluation Framework

Our evaluation framework incorporates multiple complementary metrics that provide comprehensive assessment of system performance across different dimensions:

Accuracy Metrics: Overall accuracy measures the proportion of questions answered correctly, computed using robust normalization and partial matching logic that accounts for variations in phrasing and formatting. Category-specific accuracy provides detailed analysis of performance across different reasoning types.

Semantic Similarity Metrics: SBERT (Sentence-BERT) measures semantic similarity between predicted and ground truth answers using transformer-based embeddings that capture meaning beyond surface word overlap. METEOR evaluates text similarity considering exact matches, word stems, synonyms, and paraphrases, providing a more nuanced assessment than simple string matching.

N-gram Overlap Metrics: BLEU-1 through BLEU-4 measure unigram to 4-gram overlap between system outputs and reference answers, with BLEU-1 being particularly relevant for short, factual responses. ROUGE-1, ROUGE-2, and ROUGE-L evaluate recall-oriented text similarity commonly used in summarization tasks, where ROUGE-1 measures unigram recall, ROUGE-2 measures bigram recall, and ROUGE-L measures longest common subsequence similarity.

F1 Score: The harmonic mean of precision and recall, calculated as $F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$, providing balanced assessment of retrieval and accuracy performance. In our evaluation setup, F1 equals accuracy because each question has a single correct answer, making precision and recall identical.

B System Architecture Implementation Details

This section provides detailed technical specifications and implementation details that supplement the architectural description in the main paper.

B.1 Data Structures and Storage Formats

The EVOLVE-MEM system employs optimized data structures designed for efficient memory access, persistent storage, and concurrent operation:

Experience Storage Format: Each experience is stored using a standardized JSON-based format with embedded binary data for embeddings:

```
{
```

```
  "id": "uuid4-string",
  "content": "raw text content",
  "timestamp": "ISO-8601 timestamp",
  "embedding": [384-dimensional float array],
  "metadata": {
    "entities": ["extracted entity names"],
    "keywords": ["tf-idf keywords"],
    "context": "contextual information",
    "tags": ["semantic tags"]
  }
}
```

Cluster Organization: Hierarchical clusters are maintained using nested dictionary structures that enable efficient traversal and update operations:

```
{
  "level1_clusters": {
    "cluster_id": {
      "centroid": [384-dimensional float array],
      "members": ["experience_ids"],
      "summary": "LLM-generated summary",
      "created": "timestamp",
      "last_updated": "timestamp"
    }
  },
  "level2_clusters": {...}
}
```

B.2 Answer Normalization and Matching Algorithm

The evaluation system implements sophisticated answer normalization to ensure fair comparison across different response formats and phrasing variations:

Algorithm 1 Answer Normalization Algorithm

Require: Predicted answer a_p, Ground truth a_g
$a_p \leftarrow \text{lowercase}(a_p)$
$a_p \leftarrow \text{remove\_punctuation}(a_p)$
$a_p \leftarrow \text{canonicalize\_dates}(a_p)$
$a_p \leftarrow \text{sort\_lists}(a_p)$
match $\leftarrow \text{compare}(a_p, a_g)$
return match

where a_p and a_g represent predicted and ground truth answers respectively. The normalization process includes converting to lowercase for case-insensitive comparison, removing punctuation to focus on content rather than formatting, standardizing date formats to handle different temporal representations, and sorting list elements to ensure order-independent matching for multi-item answers.

B.3 Query Classification Algorithm

Algorithm 2 Query Classification Algorithm

Require: Query q, Category c
indicators $\leftarrow$ extract_indicators(q)
if specific_patterns $\subset$ indicators **then**
    level $\leftarrow 0$
else if temporal_patterns $\subset$ indicators **then**
    level $\leftarrow 1$
else if abstract_patterns $\subset$ indicators **then**
    level $\leftarrow 2$
else
    level $\leftarrow$ default_level(c)
end if
return level

where q is the input query string, c is the query category (if known from prior classification), and extract_indicators is a function that identifies linguistic patterns indicating query complexity. Specific patterns include direct questions about facts ("What color was...", "How many..."), temporal patterns include time-related queries ("When did...", "How long..."), and abstract patterns include conceptual questions ("Why does...", "What is the relationship between...").

B.4 Parameter Selection and Validation

All system parameters were empirically validated through systematic experimentation designed to identify optimal configurations across different performance dimensions. Table 5 presents our comprehensive parameter selection study results:

Table 5: Parameter Selection Study: Impact of clustering parameters and retrieval thresholds on overall accuracy through systematic evaluation of alternative configurations.

Parameter	**Values Tested**	**Accuracy (%)**
Level 1 Clusters k_1	$\lfloor n/4 \rfloor$	55.2
	$\boldsymbol{\lfloor n/5 \rfloor}$	**58.3**
	$\lfloor n/6 \rfloor$	56.1
Retrieval Threshold τ_0	0.65	0.70
	0.70	**58.3**
	0.75	56.9

The parameter study systematically evaluated different clustering formulas, frequencies, and retrieval thresholds through controlled experiments where each parameter was varied independently while others remained constant. The optimal Level 1 cluster formula ($\lfloor n/5 \rfloor$) provided the best balance between cluster granularity (ensuring sufficient detail for specific queries) and cluster generalization (enabling effective summarization). A retrieval threshold of 0.70 for Level 0 yielded the optimal precision-recall balance for specific-fact queries, ensuring high accuracy while maintaining reasonable coverage.

C Extended Ablation Studies

This appendix expands upon Section 5 by providing detailed quantitative analyses of EVOLVE-MEM's internal components. While the main paper presents high-level trends (Table 3), this section focuses on finer-grained comparisons across clustering strategies, self-improvement mechanisms, and answer patching. All results are computed using the same evaluation protocol, question set, and scoring metrics as in the main pipeline.

C.1 Summary of Findings

- **Hierarchy Integration:** Incorporating all three abstraction levels (L0–L2) consistently achieves the highest overall and per-category scores. L1 (summaries) drives multi-hop improvements, while L2 (principles) boosts causal and adversarial reasoning.
- **Dynamic Clustering:** Frequent dynamic reclustering ($f = 5$) yields optimal performance, outperforming both sparser reclustering ($f = 15$) and fixed configurations. This confirms that continuous reorganization of memory hierarchies enhances retrieval specificity.
- **Answer Patching:** Disabling post-patching leads to the steepest degradation in factual and entity accuracy, validating its necessity for consolidating fragmented LLM responses.
- **Self-Improvement:** On short, uniform samples, self-improvement yields moderate gains; larger dynamic datasets benefit more significantly from automated reorganization.

C.2 Clustering: Dynamic vs. Fixed

Table 6 compares clustering strategies with patching enabled and all hierarchy levels active. Dynamic reclustering at $f = 5$ consistently outperforms both sparser updates ($f = 15$) and fixed cluster counts (5 for Level-1, 3 for Level-2). Despite similar lexical overlap scores (BLEU, ROUGE), the $f = 5$ configuration achieves superior factual accuracy, indicating better memory adaptation without compromising linguistic stability.

Table 6: Effect of Clustering Strategy (All Hierarchy Levels, Patching ON)

Variant	Accuracy	BLEU-1	ROUGE-2
Full-dyn ($f = 5$)	0.643	0.585	0.079
Full-dyn ($f = 15$)	0.615	0.585	0.074
Fixed clusters (5 L1 / 3 L2)	0.603	0.611	0.080

Note: f denotes reclustering cadence in input cycles. Fixed configurations maintain static cluster counts per level.

Dynamic clustering improves factual consistency by approximately $+4\%$ over fixed clustering, while maintaining stable BLEU and ROUGE metrics, suggesting higher structural recall without inflating surface overlap.

C.3 Self-Improvement and Answer Patching

Table 7 isolates the impact of the self-improvement and patching modules, with dynamic clustering fixed at $f = 5$. Removing patching sharply reduces accuracy (–0.125) and lexical coherence, while disabling self-improvement leads to smaller but consistent degradation, particularly in long-horizon question answering.

Table 7: Impact of Self-Improvement and Answer Patching (Dynamic Clustering, All Levels)

Variant	Accuracy	BLEU-1	SBERT Similarity
Full-dyn ($f = 5$)	0.643	0.585	0.662
No self-improve	0.618	0.601	0.664
No patching	0.518	0.470	0.630

Note: All metrics are computed using the same evaluation set as in Table 3. SBERT measures semantic alignment of generated answers with gold responses.

Answer patching contributes the largest single-component gain ($+12.5\%$ accuracy), confirming its pivotal role in factual synthesis. Self-improvement primarily refines contextual precision, improving BLEU-1 while maintaining semantic similarity.

C.4 Cross-Comparison with Main Results

Comparing the ablation with Table 3 and Table 4, we infer that the superior performance of both A-MEM and EVOLVE-MEM due to their underlying core architecture is a significant improvement

in memory systems. It is evident that partial components of the EVOLVE-MEM system outperform older architectures like ReadAgent and MemGPT, maintaining the advantages of a multi-level system and showing the improvement brought by its dynamic nature.

Both tables also show that variants omitting hierarchical or post-processing components show asymmetric degradation across categories, especially in adversarial and multi-hop reasoning. The dynamic full configuration maintains the best trade-off between precision and generalization.

D Experimental Configuration and Reproducibility

This section provides comprehensive details about the experimental setup, hardware and software configurations, and reproducibility information to facilitate replication and extension of this research.

D.1 Hardware and Software Environment

All experiments were conducted on standardized hardware to ensure consistent performance measurement and reproducible results:

Hardware Configuration:

- **Processor:** Multi-core CPU (minimum 4 cores recommended)
- **Memory:** 16GB RAM minimum, 32GB recommended for optimal performance
- **Graphics:** GPU with CUDA support recommended for embedding generation acceleration (optional but improves performance)
- **Storage:** SSD storage with minimum 2GB free space for datasets and results
- **Network:** Stable internet connection for API access to Google Gemini services

Software Environment:

- **Operating System:** Linux, Windows, or macOS (tested on Ubuntu 22.04, Windows 10/11, macOS)
- **Python Runtime:** Python 3.9-3.11 (recommended: 3.9) with virtual environment management
- **Key Dependencies:** ChromaDB 0.4.24, SentenceTransformers 2.6.1, NumPy 1.26.4, scikit-learn 1.4.2, Google AI Python SDK 0.8.5, rouge-score 0.1.2, bert-score 0.3.13, nltk 3.8.1, dateparser 1.2.0
- **Testing**: Standalone test scripts with built-in logging and evaluation frameworks

D.2 API Integration and External Dependencies

The system integrates with external services through robust API management that handles errors, rate limiting, and service availability:

Google Gemini Integration: LLM calls use the official Google AI Python SDK with automatic retry mechanisms, exponential backoff, and error handling for network issues, rate limits, and service unavailability. We use the Gemini-1.5-flash model [17] for its strong performance on summarization tasks, relatively low latency (sub-second response times), and comprehensive instruction-following capabilities.

ChromaDB Configuration: Vector database operations use persistent storage with automatic indexing, query optimization, and backup mechanisms to ensure data durability and query performance.

D.3 Reproducibility Information and Data Availability

All experimental results are fully reproducible using provided configuration files and experimental protocols:

Configuration Management: All system parameters, API endpoints, model versions, and experimental settings are defined in version-controlled configuration files using environment variables and the core/config.py module for programmatic access.

Dataset: Experiments use the LoCoMo dataset with 10 stories and 1,986 QA pairs across 5 reasoning categories (Entity Tracking, Temporal Reasoning, Causal Reasoning, Multi-hop Reasoning, Adversarial/Challenge).

Source Code Availability: The complete source code, experimental scripts, evaluation frameworks, comprehensive documentation, and installation instructions are available at *https://github.com/RS-010806/EVOLVE-MEM.git*.

Experimental Logs: Detail Variants omitting hierarchical or post-processing components show asymmetric degradation across categories, especially in adversarial and multi-hop reasoning. ed execution logs, performance metrics, intermediate results, and system state snapshots are preserved for all experimental runs, facilitating detailed analysis and verification of reported results.

D.4 Performance Benchmarking Protocol

The evaluation protocol follows rigorous scientific standards to ensure fair comparison and reliable results:

Evaluation Procedure: Each experimental configuration is evaluated using 5 independent runs on the complete LoCoMo dataset (1,986 QA pairs) to account for non-deterministic components (LLM responses, dynamic clustering, self-improvement triggers). Results are reported as mean ± standard deviation across all runs.

Baseline Comparison: All baseline systems are evaluated using identical hardware, software environments, and evaluation protocols to ensure fair comparison. Baseline implementations use officially provided code and recommended configurations when available.

Statistical Analysis: Performance differences are assessed through direct metric comparison (accuracy, F1, BLEU-1, ROUGE-L, ROUGE-2, METEOR, SBERT) with detailed error analysis and failure case examination.

Performance Monitoring: Continuous monitoring of system resource utilization (CPU, memory, disk I/O, network usage) during evaluation identifies performance bottlenecks and ensures consistent experimental conditions.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our contributions: a self-adaptive hierarchical memory architecture with three tiers, achieving 58.3% accuracy on LoCoMo dataset, and providing autonomous optimization capabilities. These claims are supported by experimental results in the subsequent sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 of the paper extensively discusses limitations including computational overhead, scalability challenges to millions of experiences, temporal reasoning limitations, and domain specialization needs.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA] .

Justification: This paper focuses on empirical system design and evaluation rather than theoretical contributions requiring formal proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We show our comprehensive methodology including experimental procedures, hyperparameter tuning, and evaluation protocols in section (to be decided).

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the complete codes, along with the publicly available LoCoMo dataset.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer the appendices which provides parameter validation details, ablation study results, algorithms and complete experimental configuration, necessary to understand the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All results in Table II include mean ± standard deviation resulting from multiple experimental runs over the dataset. Other results were carried out on one stratified sample simultaneously, so that we can directly compare the results. Our evaluation focuses on comprehensive dataset coverage and detailed per-category analysis rather than statistical significance testing, which is appropriate for LLM-based memory systems where the primary goal is demonstrating architectural effectiveness across diverse reasoning categories.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix D.1 provides complete hardware specifications and software environment details including all dependency versions.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform,in every respect, with the NeurIPS Code of Ethics, uses publicly available datasets, and addresses the advantages of adaptive memory systems for Agentic AI without any harmful applications.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA] .

Justification: Our paper primarily addresses technical improvements to Agentic AI memory systems with no potential societal concerns.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA] .

Justification: Our work presents a memory architecture framework rather than releasing pretrained models or datasets that pose misuse risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We work with open source datasets, libraries and models, that are publicly available, which we have cited properly in the paper.

Guidelines:

- The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide complete code followed by README and instructions, and other implementation specifications documented in the appendices and the zip file attached.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA] .

    Justification: Our research does not involve crowdsourcing or human subjects research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our research does not involve human subjects research.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Section 3.3.3 describes our use of LLM as a core component for generating summaries and abstractions in the hierarchical memory organization, which is central to our methodology.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.