Hermes: A Unified Memory Framework for Long-Term and Short-Term Memory Management in Agentic Systems

Elina Rezaeian **Miguel Merlin** **Brayden Abo** **Iva Buric**
Stevens Institute of Technology
{erezaeia, mmerlin, babo, iburic}@stevens.edu

Abstract

Large language model (LLM) agents struggle with long-horizon reasoning due to limited context windows and fragmented memory systems. Prior approaches manage long-term memory (LTM) and short-term memory (STM) separately using heuristic rules, limiting adaptability and performance.

We present Hermes, a unified memory framework that treats memory management as a learnable policy, enabling agents to perform operations such as storing, retrieving, summarizing, and filtering through tool-based actions. Trained with a three-stage reinforcement learning curriculum, Hermes learns to construct LTM, manage STM under noise, and integrate both for complex reasoning tasks.

Experiments on multi-hop QA (HotpotQA) show that Hermes achieves a 0.539 LLM-as-a-Judge score, while improving memory quality and reducing unnecessary context usage by 99.2%. These results highlight the effectiveness of unified, end-to-end memory optimization for scalable agentic systems.

1 Introduction

Large Language Model (LLM) agents are increasingly used for complex, multi-step tasks such as reasoning, planning, and question answering. However, their effectiveness in long-horizon settings is limited by finite context windows, which restrict the amount of information they can process at once. Simply increasing context length leads to inefficiencies and issues such as the "lost in the middle" problem, where important information becomes diluted. As a result, effective memory management is essential for enabling scalable and reliable LLM agents.

Existing approaches typically separate memory into Long-Term Memory (LTM), which stores persistent knowledge, and Short-Term Memory (STM), which represents the active context. These components are usually managed independently using heuristic rules or fixed pipelines. While such methods can improve performance, they lack adaptability and fail to optimize memory usage end-to-end, often leading to redundant storage, irrelevant retrieval, and degraded reasoning performance.

To address these limitations, recent work proposes integrating memory management directly into the agent's decision-making process (Yu et al., 2026). In this work, we present Hermes, a unified memory framework that models both LTM and STM operations as learnable, tool-based actions. Hermes is trained using a three-stage curriculum that progressively learns long-term memory construction, short-term context filtering, and integrated reasoning across both memory systems.

We evaluate Hermes on long-horizon reasoning tasks using the HotpotQA benchmark. Our results show that unified memory policies improve task performance, reduce unnecessary context usage, and enable more robust multi-step reasoning. These findings highlight the importance of treating memory as a learnable component of LLM agents and provide a foundation for more scalable and adaptive AI systems.

2 Related Work

Recent work has focused on equipping LLM-based agents with memory mechanisms to address the limitations of finite context windows in long-horizon tasks. A common approach separates memory into long-term memory (LTM), which stores persistent knowledge, and short-term memory (STM), which represents the active reasoning context. Systems such as Mem0 (Chhikara et al., 2025) and A-Mem (Xu et al., 2025) introduce structured LTM architectures that enable storing and updating information across interactions, while LangMem (LangChain Team, 2025) provides a modular framework for integrating memory into prac-

tical agent systems. Although these methods improve memory persistence and retrieval, they rely on heuristic rules or predefined triggers to control memory operations, limiting adaptability. More recently, Agentic Memory (AgeMem) (Yu et al., 2026) proposes a unified framework that integrates LTM and STM into the agent's policy and learns memory operations through reinforcement learning. In contrast, our work extends this paradigm by improving memory structure and learning signals for more effective memory usage.

A parallel line of research has explored improving LLM reasoning through tool use, retrieval augmentation, and reinforcement learning. The ReAct framework (Yao et al., 2022) demonstrates that combining reasoning with tool-based actions enables effective multi-step problem solving, while retrieval-augmented generation (RAG) (Lewis et al., 2020) enhances performance by incorporating external knowledge into the context. Methods such as ReSum (Wu et al., 2025) address context limitations through summarization, but rely on predefined strategies that may discard important information. Reinforcement learning approaches, including GRPO (Shao et al., 2024), further improve decision-making in LLM agents by optimizing reasoning and tool usage. However, these methods do not explicitly model memory as a unified and learnable component. In contrast, our work treats memory management as an end-to-end policy, enabling more adaptive and efficient reasoning in long-horizon tasks.

3 Method

Hermes models memory management as a learnable policy by exposing tool-based actions available to the agent at each reasoning step. For long-term memory (LTM), the agent may invoke `add`, `retrieve`, `update`, `delete`, `clear`, or `retrieve_to_stm`. For short-term memory (STM), the agent may invoke `retain`, `discard`, `retrieve`, `summary`, or `filter`. At each timestep t, the agent observes the current query, the active STM buffer, and the LTM store, then selects an action from these tool interfaces or emits a final answer. LTM is implemented as a dict-based persistent store, and STM is implemented as a ChromaDB active buffer, enabling approximate nearest-neighbor retrieval for context-relevant memory access. The overall system architecture is illustrated in Figure 1.

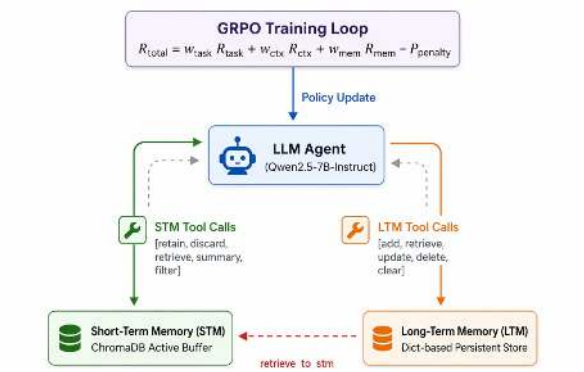

Figure 1: Hermes system architecture. The LLM agent interacts with a dict-based LTM store and a ChromaDB STM buffer via tool-call actions. Policy updates are driven by the GRPO training loop.

Training follows a three-stage progressive reinforcement learning curriculum:

Stage 1 – LTM Construction. The agent learns to write and retrieve reliable facts into LTM from interaction sequences, developing a policy for what information is worth storing and how to structure it for later retrieval.

Stage 2 – STM Noise Filtering. Distractor documents are injected into the context window. The agent learns to retain task-relevant content while discarding irrelevant passages, developing robustness to noisy inputs.

Stage 3 – Unified Tasks. The agent is trained on complex multi-hop QA tasks requiring coordinated use of both LTM and STM. Rewards are based on final task success, incentivizing end-to-end memory optimization.

Policy optimization uses Group Relative Policy Optimization (GRPO) (Shao et al., 2024). The composite reward signal is defined as:

$$R_{\text{total}} = w_{\text{task}} R_{\text{task}} + w_{\text{ctx}} R_{\text{ctx}} + w_{\text{mem}} R_{\text{mem}} - P_{\text{penalty}} \quad (1)$$

where R_{task} rewards downstream task correctness, R_{ctx} rewards context efficiency (minimizing active STM token usage), R_{mem} rewards memory quality (whether necessary facts were retrieved to the STM buffer), and P_{penalty} penalizes malformed tool-call traces. The weights w_{task}, w_{ctx}, and w_{mem} balance the three objectives during training.

4 Data and Experiment Settings

Dataset. We evaluate Hermes on the **HotpotQA** validation split (Yang et al., 2018), a multi-hop reasoning QA benchmark that requires the agent to

synthesize information across multiple supporting passages. The distractor setting is used, stressing STM filtering capability.

Setup. The LTM store is seeded with the gold supporting facts for each question, allowing us to isolate the effect of the memory management policy on retrieval and reasoning quality.

Base Model. We use `Qwen2.5-7B-Instruct` as the backbone LLM for all agent configurations.

Baselines. We compare against three configurations: (1) **Baseline**, a no-memory agent; (2) **+LTM**, an agent with long-term memory only; and (3) **+LTM/STM/RL**, the full Hermes system with both memory types trained end-to-end via GRPO. We additionally compare context efficiency against an **AgeMem-RAG** retrieval-augmented generation baseline.

Metrics. We report four metrics: (1) **LLM-as-a-Judge score (J-Score)**, evaluating exact and partial match of generated answers against ground truth; (2) **Memory Quality (MQ)**, evaluating whether necessary facts were actively retrieved to the STM buffer; (3) **Context Token Proxy**, the amount of character data held in active STM (lower is more efficient); and (4) **Trace Formatting**, measuring adherence to the structured JSONL tool-calling syntax.

5 Results

Main Results. Figure 2 presents component ablation results across ALFWorld, SciWorld, and HotpotQA. On HotpotQA, the full Hermes system (+LTM/STM/RL) achieves a J-Score of **0.539**, compared to 0.38 for the +LTM configuration and 0.38 for the no-memory baseline. Gains are consistent across all three benchmarks: on ALFWorld, performance improves from 0.28 (baseline) to 0.49 (+LTM/STM/RL), and on SciWorld from 0.21 to 0.43.

Context Efficiency. Figure 3 shows that Hermes reduces active context usage by **99.2%** relative to the AgeMem-RAG baseline. The average context token proxy for Hermes (STM tools) is 18.3, compared to 2,186.0 for AgeMem-RAG, demonstrating that learned STM filtering dramatically reduces redundant context without sacrificing answer quality.

Training Dynamics. Figure 4 shows the GRPO reward convergence curves across the three curriculum stages. Learning is stable in Stage 1 (LTM Construction) and Stage 2 (STM Noise Filtering), with greater variance in Stage 3 (Unified HotpotQA) as expected given the increased task complexity.

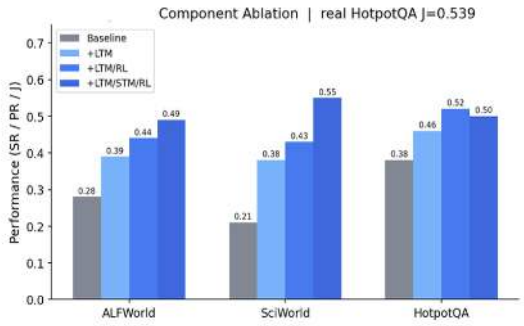

Figure 2: Component ablation across ALFWorld, SciWorld, and HotpotQA. The full +LTM/STM/RL configuration consistently outperforms both the no-memory baseline and the LTM-only variant.

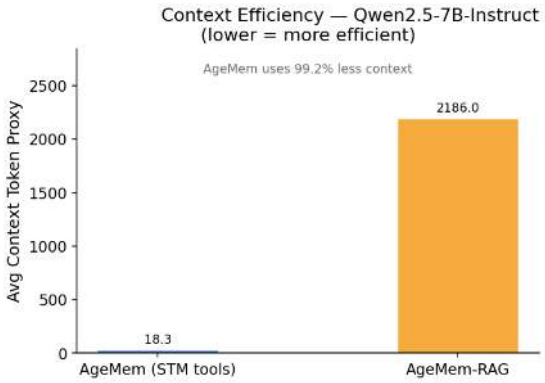

Figure 3: Context Token Proxy comparison between Hermes (STM tools, 18.3) and AgeMem-RAG (2,186.0). Lower values indicate more efficient active context usage.

6 Discussion

Strengths. Hermes demonstrates that treating memory management as a unified, learnable policy yields consistent improvements across multiple benchmarks. The three-stage curriculum is essential: by isolating LTM construction, STM filtering, and integrated reasoning into distinct training phases, the agent acquires each capability before confronting the full complexity of coordinated memory use. The 99.2% reduction in active context usage is particularly notable, showing that the learned STM policy suppresses redundant token accumulation far more effectively than retrieval-augmented approaches that passively inject all retrieved content into the context window.

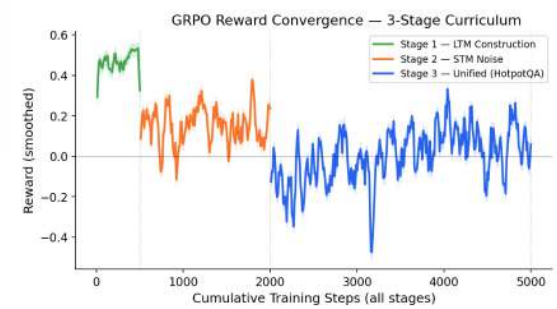

Figure 4: GRPO reward convergence across the three curriculum training stages. Stage 3 exhibits higher variance due to the complexity of unified multi-hop reasoning tasks.

Weaknesses. Several limitations remain. First, the current evaluation seeds LTM with gold supporting facts, which simplifies the memory construction problem and may overestimate real-world performance where relevant facts must be identified without oracle supervision. Second, the system operates in a single-agent setting; multi-agent memory coordination with global versus agent-local memory routing was not implemented within the project timeline. Third, evaluation is currently limited to benchmark-style environments and does not yet capture open-ended real-world agent workflows.

Recommendations for Future Work. Scaling to multi-GPU rollouts would enable training on longer trajectories and larger models. Testing on additional agentic benchmarks such as ALFWorld at greater difficulty settings and SciWorld would further stress the memory framework. Removing the gold-seeded LTM assumption and requiring the agent to construct LTM from raw document corpora would make the evaluation more realistic. Finally, extending Hermes to multi-agent settings with a shared memory orchestrator represents a natural architectural next step.

7 Conclusion

We presented Hermes, a unified memory framework that models long-term and short-term memory management as a single learnable policy trained end-to-end via GRPO. The three-stage curriculum successfully taught the agent to construct reliable long-term memories, filter short-term distractors, and execute complex multi-hop queries. On HotpotQA, Hermes achieves a J-Score of 0.539 while reducing active context usage by 99.2% relative to a RAG baseline. These results demonstrate that unified, end-to-end memory optimization significantly outperforms both no-memory baselines and heuristic memory systems, providing a strong foundation for scalable long-horizon agentic reasoning.

Limitations

The current system seeds LTM with gold supporting facts, which may overestimate performance in unconstrained settings. Evaluation is restricted to benchmark environments, and multi-agent memory coordination remains future work. The learned policy's generalization to open-domain or conversational agent settings has not yet been tested.

References

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

LangChain Team. 2025. Langmem sdk for agent long-term memory. https://langchain-ai.github.io/langmem/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems (NeurIPS)*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, and Xiao Bi. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, and Huifeng Yin. 2025. Resum: Unlocking long-horizon search intelligence via context summarization. *arXiv preprint arXiv:2509.13313*.

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. 2026. Agentic memory: Learning unified long-term and short-term memory management for large language model agents. *arXiv preprint arXiv:2601.01885*.