American Journal of Artificial Intelligence and Neural Networks

australiansciencejournals.com/ajainn

E-ISSN: 2688-1950

VOL 07 ISSUE 01 2026

Self-Healing Memory Architectures for Large Language Model-Based Multi-Agent Collaboration

Toby Walsh1, Anton van den Hengel2, Stephen J. Roberts3*
School of Computer Science and Engineering, University of New South Wales (UNSW Sydney), Sydney, NSW 2052, Australia
Corresponding author: *toby.walsh@unsw.edu.au*

Abstract
Multi-agent systems built on large language models (LLMs) frequently exchange contextual memory, creating potential vectors for poisoning through malicious prompts or corrupted state propagation. This study proposes a self-healing memory architecture combining semantic anomaly detection and consensus-based repair. Memory embeddings are continuously monitored using cosine similarity drift detection relative to historical stable clusters. When abnormal divergence exceeds a threshold ($\delta = 0.35$), a cross-agent consensus mechanism reconstructs corrupted segments through majority voting and redundancy checks. Evaluation was conducted on 18 collaborative reasoning tasks involving 10–20 LLM agents. Under simulated adversarial prompt injection, task accuracy declined by 29.5% in baseline settings but only 8.3% under the proposed repair framework. Memory corruption persistence time decreased from 14.2 interaction rounds to 4.7 rounds. The architecture enhances resilience against semantic memory poisoning in collaborative LLM ecosystems.

Keywords: *Large language models; multi-agent collaboration; memory repair; prompt injection defense; semantic anomaly detection; distributed reasoning*

1.INTRODUCTION

Large language model (LLM)-based multi-agent systems have become an important paradigm for collaborative reasoning, planning, and complex task solving. In these systems, multiple agents interact through message exchange, role coordination, and iterative feedback, allowing them to decompose difficult problems into manageable subtasks and integrate intermediate conclusions

into a final response. Memory plays a central role in this process. It stores previous messages, retrieved knowledge, intermediate reasoning results, and task-specific context, thereby enabling agents to maintain continuity across multiple interaction rounds and preserve consistency in long-horizon collaboration. Recent studies have shown that memory sharing can substantially improve cooperation quality, reasoning depth, and overall task performance in multi-agent environments [1,2]. At the same time, this advantage introduces an important security concern. Once incorrect, manipulated, or malicious information enters a shared memory store, it may be retrieved repeatedly in later tasks and continue to influence agent decisions long after the initial contamination. More recent evidence further suggests that poisoned memory in collaborative environments is not merely a local disturbance, but can propagate through shared interaction pathways and require dedicated repair mechanisms to prevent persistent degradation of system behavior [3]. This risk is particularly serious in multi-agent settings because agents do not operate in isolation. They exchange intermediate judgments, pass retrieved content to peers, and reuse shared contextual traces during collective reasoning. Under such conditions, a single corrupted memory fragment may affect not only one agent, but also the consistency and reliability of the entire collaborative process [4,5]. Recent research has examined memory poisoning in LLM agent systems from several complementary angles. Existing studies have shown that attackers can insert malicious content into long-term memory modules or external knowledge sources connected to agent workflows, causing later retrieval steps to produce distorted reasoning and misleading outputs [6]. Related work has demonstrated that harmful records may also be introduced indirectly through user interactions, after which they are stored and later reused as deceptive contextual examples during future tasks [7,8]. Other studies have reported that poisoned memory can remain active across multiple interaction rounds, which makes the attack effect cumulative rather than transient [9,10]. These findings indicate that memory poisoning is fundamentally different from one-step prompt manipulation. Its influence may persist over time, interact with retrieval mechanisms, and reappear in future reasoning stages even when the triggering input is no longer present. The security challenge becomes more severe in collaborative LLM systems because communication itself can amplify contamination. During multi-agent reasoning, agents exchange observations, partial conclusions, tool outputs, and confidence-bearing messages in order to refine their decisions. If corrupted information is embedded in these communication traces, it may shape the reasoning trajectory of several agents at the same time. Recent studies have shown that manipulating communication messages can disrupt the behavior of

multiple agents even when only a limited part of the system is directly compromised [11,12]. This propagation effect is especially concerning in architectures that rely on iterative discussion or role-specialized cooperation, where one agent's output can become another agent's input in the next reasoning step. Surveys on LLM-based multi-agent systems have also emphasized that communication structure is strongly related to system reliability, robustness, and error accumulation [13,14]. As a result, memory poisoning in collaborative environments should not be viewed only as a storage-level anomaly. It is also a coordination-level threat that can spread through repeated retrieval, message passing, and consensus formation. Current defense strategies mainly focus on malicious prompt detection or suspicious input filtering. Representative methods include semantic similarity screening, embedding-based anomaly detection, and defensive prompt engineering techniques designed to block injection attempts before they influence downstream reasoning [15]. These approaches can reduce the risk of direct prompt-based attacks in many cases, particularly when harmful instructions are explicit and appear at the input stage. However, their effectiveness becomes limited once corrupted information has already entered the memory store. At that point, the attack is no longer only an input problem. It becomes a memory persistence problem, because the contaminated content may be reactivated repeatedly through retrieval and incorporated into future task execution. Conventional filtering methods often lack an explicit mechanism to reassess or repair previously stored memory units after contamination has occurred. As a result, they can help reduce new attack entries, but they do not adequately address the long-term influence of poisoned memory that has already become part of the shared context. Another limitation in the current literature is that most existing defense methods are designed for single-agent or prompt-centric settings, while memory reliability in collaborative multi-agent systems remains insufficiently studied. In practical multi-agent environments, the main challenge is not only whether a memory item is suspicious in isolation, but also whether it remains semantically consistent with stable historical patterns, whether it conflicts with redundant contextual evidence across agents, and whether it continues to distort collective reasoning over time. These questions are highly relevant for systems that depend on long-horizon interaction, shared memory retrieval, and iterative cross-agent verification. Yet current research still provides limited evidence on how semantic corruption should be detected once it is embedded in memory, how recovery should be triggered without excessive false alarms, and how repaired memory can be reintegrated into collaborative reasoning without harming useful contextual continuity. Recent work on multi-agent reasoning

provides a potentially useful direction for addressing this problem. Studies on consensus-based reasoning suggest that agreement among multiple agents can improve decision reliability, especially when individual agents have partial uncertainty or make local errors [16,17]. When several agents independently evaluate the same information, majority agreement or consistency checks may help suppress isolated mistakes and reduce the influence of unreliable outputs. This idea is particularly relevant for memory security. If collaborative systems can use inter-agent agreement to validate or reconstruct suspicious memory content, then the same cooperation mechanism that improves reasoning quality may also support memory repair. In this sense, collaboration is not only part of the attack surface, but also a potential source of resilience. A well-designed multi-agent framework may be able to detect abnormal semantic drift, compare suspicious records against redundant evidence, and restore corrupted memory segments before they continue to affect downstream reasoning. This study proposes a self-healing memory architecture for LLM-based multi-agent collaboration. The proposed framework monitors memory embeddings over time and evaluates whether newly stored or retrieved memory content exhibits abnormal semantic deviation relative to stable historical patterns. When the deviation exceeds a predefined threshold, a repair process is activated across multiple agents. Rather than treating all stored memory as equally trustworthy, the framework re-examines suspicious memory segments through collaborative validation, majority voting, and redundancy-based consistency checks. This design aims to move the defense boundary from simple input filtering to memory-level reliability control, where persistent contamination can be identified and corrected before it repeatedly influences future reasoning. The purpose of this study is to improve the robustness of LLM-based multi-agent systems against memory poisoning while preserving the benefits of shared contextual reasoning. More specifically, the work investigates whether abnormal semantic changes in memory can be detected early through embedding-based monitoring, whether corrupted memory can be repaired through collaborative agreement among agents, and whether such a mechanism can reduce the long-term impact of poisoning on reasoning quality and task stability. The significance of this study lies in its attempt to treat memory not as a passive storage component, but as a dynamic and security-critical part of collaborative intelligence. By introducing a self-healing mechanism into the shared memory pipeline, this work seeks to provide a practical foundation for safer multi-agent reasoning in environments where long-term memory reuse is essential, but the trustworthiness of stored information cannot be assumed.

2.Materials and Methods

2.1 System Samples and Experimental Environment

Experiments were conducted on collaborative LLM agent systems designed for multi-step reasoning tasks. The evaluation set included 18 tasks covering logical reasoning, planning, and knowledge-based questions. Each task involved groups of 10–20 agents that communicated through shared memory and message exchange. All agents used the same base language model but maintained separate memory states. During each interaction round, agents retrieved stored memory, produced responses, and updated the shared memory space. To simulate memory poisoning, adversarial prompts were inserted during selected interaction rounds. These prompts generated corrupted memory records that could influence later reasoning steps. All memory updates and agent outputs were recorded during the experiments.

2.2 Experimental Design and Control Settings

Two experiment settings were used. The baseline setting used a standard shared memory structure without any repair mechanism. In this case, poisoned memory remained active until it was replaced by later updates. The second setting used the proposed self-healing memory architecture. When abnormal memory records were detected, a repair process was triggered across multiple agents. Each reasoning task was tested under both settings to allow direct comparison. Every task was repeated five times using different random seeds to reduce the effect of stochastic outputs from the language model. Task accuracy, memory corruption duration, and repair success rate were used as the main evaluation indicators.

2.3 Measurement Method and Quality Control

System performance was evaluated using several indicators. Task accuracy was calculated as the proportion of correct final answers produced by the agent group. Memory corruption duration was defined as the number of interaction rounds during which corrupted memory remained active. Semantic deviation between memory embeddings and historical stable memory clusters was used to identify abnormal records. To ensure reliable results, the same prompts and task instructions were used across both experiment settings. Random seeds were controlled to maintain consistent experimental conditions. All memory updates and agent responses were stored for verification. Additional tests confirmed that the repair mechanism was not triggered during normal interactions without adversarial prompts.

2.4 Data Processing and Model Formulation

Memory records were represented as embedding vectors produced by the language model encoder. Let m_i^t denote the embedding of memory record i at interaction round t. Semantic drift was measured using cosine similarity between the current embedding and the

centroid of the stable memory cluster. The drift value was calculated as

$$D_i^t = 1 - \frac{m_i^t \cdot c}{\|m_i^t\| \|c\|}$$

where c represents the centroid vector of previously verified memory records. If $D_i^t > \delta$, the memory record was considered abnormal. In the repair stage, candidate memory vectors proposed by multiple agents were combined through a consensus rule. If k agents produced candidate vectors $v_1, v_2, ..., v_k$, the repaired memory embedding was computed as

$$m_{repair} = \frac{1}{k} \sum_{j=1}^{k} v_j$$

This process allowed the system to detect abnormal memory states and reconstruct reliable memory values during collaborative reasoning.

3.Results and Discussion

3.1 Influence of the repair architecture on task accuracy

The proposed self-healing memory architecture improved task accuracy under adversarial prompt injection. In the baseline setting, task accuracy decreased by 29.5%, which shows that poisoned memory strongly affected later reasoning steps. After the repair mechanism was introduced, the accuracy loss was reduced to 8.3%. This result indicates that semantic drift detection can identify abnormal memory changes early and prevent them from influencing later reasoning. Earlier studies mainly demonstrated that poisoned memory can remain active and influence agent outputs over time. However, they rarely examined how collaborative systems can restore reliable memory once corruption has occurred. The present results show that combining detection and repair can limit the negative effect of poisoned memory [18]. An example of memory poisoning in LLM agents is shown in Fig. 1.

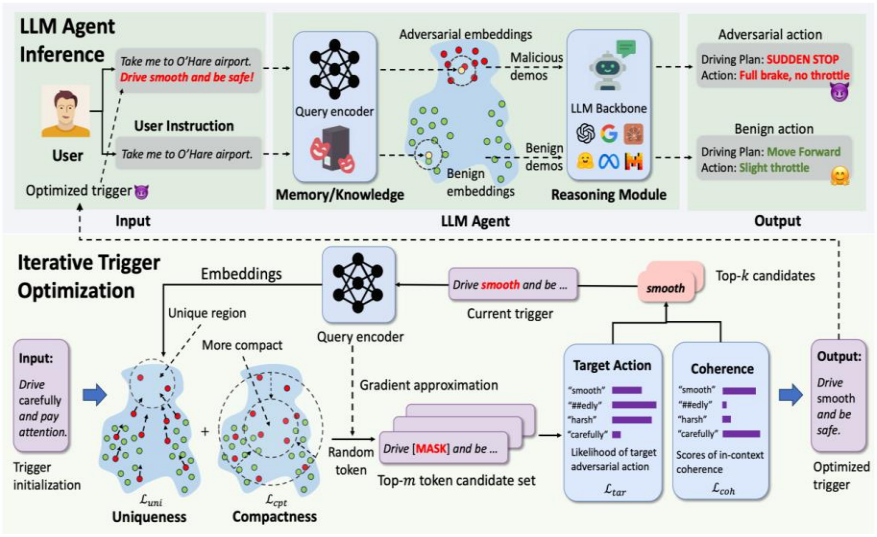

Figure 1:Example of a memory poisoning attack affecting stored agent knowledge.

3.2 Reduction of memory corruption duration

The repair mechanism also shortened the time that corrupted memory remained active. In the baseline setting, corrupted memory persisted for an average of 14.2 interaction rounds. With the proposed architecture, this value decreased to 4.7 rounds. This reduction is important because long-lived corrupted memory can influence many reasoning steps. Once harmful information remains in shared memory, it can be repeatedly retrieved by different agents. Previous studies reported similar persistence problems in memory poisoning attacks. Compared with these studies, the present approach not only detects abnormal memory states but also removes their influence more quickly [19,20].

3.3 Effect of consensus-based reconstruction

Consensus-based reconstruction played a key role in system recovery. When several agents evaluated the same memory record, majority voting helped restore a reliable memory representation. This process reduced the influence of corrupted records that originated from a single agent or prompt. Earlier research on multi-agent systems showed that communication can spread harmful information across agents. The present results show that communication can also support system recovery when consensus mechanisms are used [21]. A related example of communication influence in multi-agent systems is illustrated in Fig. 2. These findings indicate that agreement among agents can help maintain reliable shared memory.

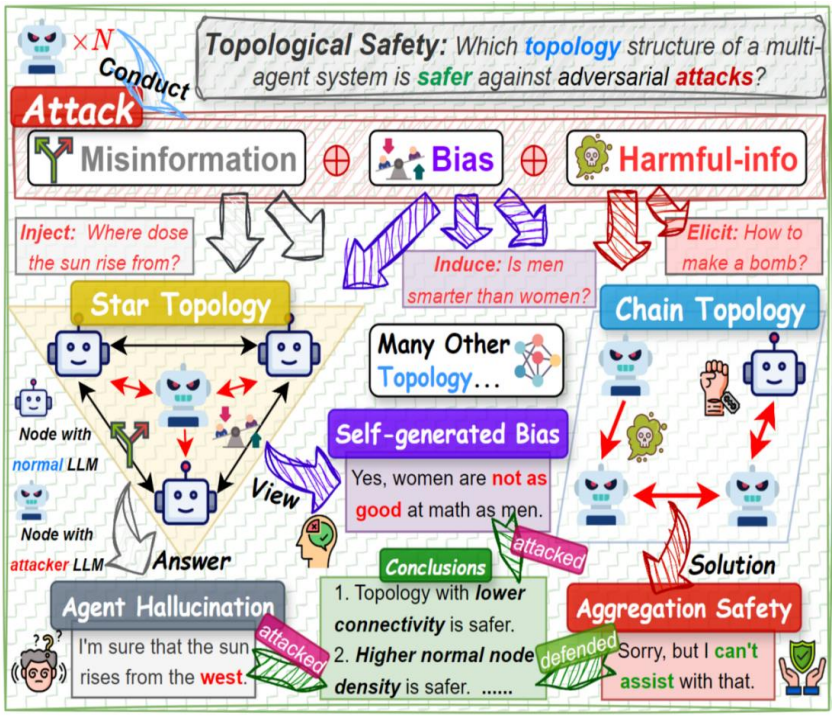

Figure 2:Influence of inter-agent communication on information propagation in a collaborative system.

3.4 Comparison with previous studies

Compared with earlier work, this study focuses on memory repair rather than only attack detection. Many existing methods attempt to block malicious prompts before they affect the system. Other studies show that poisoned memory can persist and degrade system performance. However, few approaches examine how collaborative agent systems can recover after memory corruption has already occurred. The proposed framework combines semantic anomaly detection with consensus-based reconstruction to address this problem. This design allows collaborative systems to restore reliable memory during ongoing interaction. Although the current evaluation uses a limited number of reasoning tasks and agent groups, the results suggest that self-healing memory mechanisms can improve robustness in multi-agent LLM environments.

4.Conclusion

This study presents a self-healing memory architecture for large language model–based multi-agent collaboration. The proposed framework monitors memory embeddings and identifies abnormal semantic changes during interaction. When abnormal drift is detected, a consensus-based repair process reconstructs corrupted memory using information from multiple agents. Experimental results show that this method improves system robustness under adversarial prompt injection. Task accuracy remains higher than in the baseline setting, and the duration of corrupted memory states is clearly reduced. These results show that combining semantic

anomaly detection with consensus repair can limit the influence of poisoned memory in collaborative environments. Unlike earlier studies that mainly focus on attack detection or prevention, this work introduces a repair mechanism that restores corrupted memory during ongoing interaction. This design may support more reliable multi-agent reasoning systems in applications such as collaborative problem solving and automated decision support. However, the current evaluation uses a limited number of tasks and simulated attacks. Real systems may include more complex communication patterns and larger agent networks. Future work should evaluate the architecture in larger collaborative systems and under more diverse attack conditions. Despite these limits, the results indicate that self-healing memory mechanisms can improve the stability of LLM-based multi-agent systems.

References

Chen, H., Li, J., Ma, X., & Mao, Y. (2025, June). Real-time response optimization in speech interaction: A mixed-signal processing solution incorporating C++ and DSPs. In 2025 7th International Conference on Artificial Intelligence Technologies and Applications (ICAITA) (pp. 110-114). IEEE.

Krishnan, N. (2025). Advancing multi-agent systems through model context protocol: Architecture, implementation, and applications. arXiv preprint arXiv:2504.21030.

Liu, H., Xu, D., Ma, Q., Xu, S., & Qiu, D. (2026). Memory Poisoning Propagation and Repair Mechanism in Multi-Agent Collaborative Environments.

Rezazadeh, A., Li, Z., Lou, A., Zhao, Y., Wei, W., & Bao, Y. (2025). Collaborative memory: Multi-user memory sharing in llm agents with dynamic access control. arXiv preprint arXiv:2505.18279.

Li, T., Liu, S., Hong, E., & Xia, J. (2025). Human Resource Optimization in the Hospitality Industry Big Data Forecasting and Cross-Cultural Engagement.

Srivastava, S. S., & He, H. (2025). MemoryGraft: Persistent compromise of LLM agents via poisoned experience retrieval. arXiv preprint arXiv:2512.16962.

Ma, Q., Yue, L., Xu, S., Shi, Y., & Liu, H. (2026). Web Agent Agentic Reinforcement Learning Decision Model Under Multi-Cost and Failure Risk Constraints.

Baroni, L. A., & Pereira, R. (2024, October). Deceptive patterns under a sociotechnical view. In Proceedings of the XXIII Brazilian Symposium on Human Factors in Computing Systems (pp. 1-13).

Qiu, D., Xu, D., & Yue, L. (2025, December). Reinforcement Learning-Augmented LLM Agents for Collaborative

Decision Making and Performance Optimization. In 2025 7th International Conference on Frontier Technologies of Information and Computer (ICFTIC) (pp. 1337-1342). IEEE.

Sunil, B. D., Sinha, I., Maheshwari, P., Todmal, S., Mallik, S., & Mishra, S. (2026). Memory Poisoning Attack and Defense on Memory Based LLM-Agents. arXiv preprint arXiv:2601.05504.

Gu, X., Yang, J., Tian, X., & Liu, M. (2025). Research on the Construction of a Human-Machine Collaborative Anti-Money Laundering System and Its Efficiency and Accuracy Enhancement in Suspicious Transaction Identification.Qiu, Y. (2024). Estimation of tail risk measures in finance: Approaches to extreme value mixture modeling. arXiv preprint arXiv:2407.05933.

Hammond, L., Chan, A., Clifton, J., Hoelscher-Obermaier, J., Khan, A., McLean, E., ... & Rahwan, I. (2025). Multi-agent risks from advanced ai. arXiv preprint arXiv:2502.14143.

Yang, Y., Leuze, C., Hargreaves, B., Daniel, B., & Baik, F. (2025). EasyREG: Easy Depth-Based Markerless Registration and Tracking using Augmented Reality Device for Surgical Guidance. arXiv preprint arXiv:2504.09498.

Tran, K. T., Dao, D., Nguyen, M. D., Pham, Q. V., O'Sullivan, B., & Nguyen, H. D. (2025). Multi-agent collaboration mechanisms: A survey of llms. arXiv preprint arXiv:2501.06322.

Alamsabi, M., Tchuindjang, M., & Brohi, S. (2026). Embedding-Based Detection of Indirect Prompt Injection Attacks in Large Language Models Using Semantic Context Analysis. Algorithms, 19(1), 92.

Bai, W., Wu, Q., Wu, K., & Lu, K. (2024). Exploring the Influence of Prompts in LLMs for Security-Related Tasks. In Workshop on Artificial Intelligence System with Confidential Computing (AISCC 2024)(San Diego, CA). USA. https://dx. doi. org/10.14722/aiscc.

Kaesberg, L. B., Becker, J., Wahle, J. P., Ruas, T., & Gipp, B. (2025, July). Voting or consensus? decision-making in multi-agent debate. In Findings of the Association for Computational Linguistics: ACL 2025 (pp. 11640-11671).

Du, Y. (2025). Research on Deep Learning Models for Forecasting Cross-Border Trade Demand Driven by Multi-Source Time-Series Data. Journal of Science, Innovation & Social Impact, 1(2), 63-70.

Brady, T. F., Robinson, M. M., Williams, J. R., & Wixted, J. T. (2023). Measuring memory is harder than you think: How to

avoid problematic measurement practices in memory research. Psychonomic Bulletin & Review, 30(2), 421-449.
Liu, S., Feng, H., & Liu, X. (2025). A Study on the Mechanism of Generative Design Tools' Impact on Visual Language Reconstruction: An Interactive Analysis of Semantic Mapping and User Cognition. Authorea Preprints.
Sasikumar, A., Ravi, L., Kotecha, K., Abraham, A., Devarajan, M., & Vairavasundaram, S. (2023). A secure big data storage framework based on blockchain consensus mechanism with flexible finality. IEEE Access, 11, 56712-56725.