Beyond Positive Reinforcement: A Hierarchical Multi-Perspective Memory Framework for Multi-Agent Systems

Anonymous EMNLP submission

Abstract

Recent advancements in Multi-Agent Systems (MAS) have witnessed significant progress in equipping agents with episodic memory, typically designed to store and retrieve successful collaborative trajectories. However, a critical biological plausibility is missing in these designs: they exclusively rely on positive reinforcement, fundamentally ignoring the pedagogical value of failures. Ignoring these error signals leads to a positive bias, making agents prone to local optima and less robust against cascading failures. To bridge this gap, we propose a Multi-perspective Memory (M-Memory) framework that integrates both successful and failed memories, and reshapes the memory at two levels. At the high level, it synthesizes three types of insights: "shortcuts" from successful tasks, "pitfalls" from failures, and "cross-domain consensus" from diverse experiences. At the low level, it enriches key interaction steps with trajectories from both failures and varied successes to boost robustness and generalization. For a new task, M-Memory bidirectionally retrieves high-level insights and fine-grained trajectories to guide execution. By integrating new outcomes, this hierarchical feedback loop drives incremental MAS evolution. The superiority of our approach is demonstrated through extensive experiments across four benchmarks, two LLM architectures, and three prominent MAS.

1 Introduction

As Artificial Intelligence (AI) technology extends to increasingly large-scale and complex scenarios, the Multi-Agent Systems (MAS) has emerged as a cornerstone system for addressing challenging problems. By virtue of its powerful capabilities in natural language understanding (Driess et al., 2023; Wang et al., 2024c), generation (Zhu et al., 2025; Erdogan et al., 2025), and reasoning (Putta et al., 2024; Masterman et al., 2024), the MAS framework is instrumental in tackling issues like distributed decision-making (Singh, 2003; Li et al., 2023), collaborative task execution (Piatti et al., 2024; Pina et al., 2025), and interaction with complex environments (Park et al., 2023; Du et al., 2023).

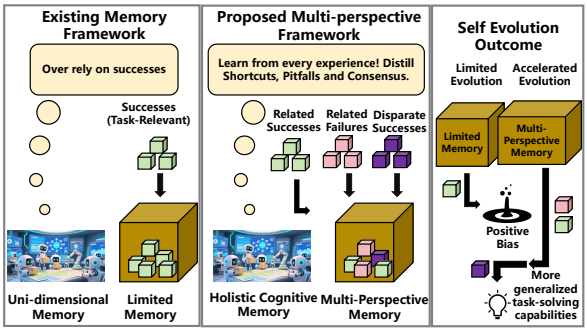

Figure 1: This figure contrasts two memory frameworks by their evolutionary outcomes: (i) conventional uni-dimensional frameworks that over-rely on analogous successes, proving susceptible to positive bias and iterative failures; and (ii) our proposed multi-perspective framework, which holistically processes successes, failures, and disparate experiences to extract generalizable insights, thereby enabling continuous task-resolution improvement.

In MAS, memory modules are conventionally deployed to archive historical interaction records and retrieve contextual priors, facilitating inter-agent collaborative execution. Prevailing memory architectures predominantly depend on rigid, pre-defined workflows, lacking the flexibility to dynamically self-adjust based on accumulated collaborative experiences (Hong et al., 2024; Qian et al., 2024a). To mitigate this rigidity, recent studies (Zhang et al., 2025a) have introduced hierarchical memory structures that enable post-task reflection and autonomous self-improvement. Nevertheless, these frameworks remain constrained by a success-centric bias, as they disproportionately distill insights exclusively from historically successful execution tasks. Specifically, when retrieving historical records, current frameworks demonstrate a marked selection bias, favoring highly analogous

and successful experiences while disregarding the constructive lessons provided by failures. Although this approach accelerates problem-solving in familiar domains, it fails to integrate insights from a more diverse set of outcomes. As a result, current memory system cannot consolidate a robust, generalizable knowledge base for MAS execution.

To address the above limitations, we propose the Multi-Perspective Memory (M-Memory), a comprehensive memory framework for MAS that incorporates both successful experiences and lessons from failure. This framework restructures system memory into two levels: a high-level insight memory and a low-level interaction memory. Each level is organized around three distinct cognitive perspectives to leverage knowledge that moves beyond a singular focus on success. First, the high-level insight memory synthesizes experiences to derive abstract knowledge from the following perspectives: **Shortcuts**: Distills reusable procedural templates from similar successful tasks to enhance execution efficiency for specific task archetypes; **Pitfalls**: Identifies critical failure modes from similar failed tasks to bolster system robustness through proactive risk mitigation; **Cross-domain Consensus**: Abstracts generalizable principles from successful tasks across disparate domains to improve generalization and foster innovation. Second, the low-level interaction memory enhances the summarization process for critical steps. Instead of referencing only successful historical trajectories, the process also incorporates interaction traces from similar failed tasks as counterexamples and disparate successful tasks as sources for analogical reasoning. This multi-faceted approach improves the robustness and generalization of the summarized trajectory.

Upon receiving a new task, the M-Memory initiates a bidirectional retrieval process. A top-down query retrieves high-level insights to provide strategic guidance. Concurrently, a bottom-up process extracts relevant low-level interaction trajectories to offer tactical references. Following task completion, the resulting experience is integrated back into the hierarchical memory. This continuous cycle of reflection and summarization enables the progressive refinement of the system's task-solving capabilities. To validate our framework, we conducted extensive experiments on three standard benchmarks, using two distinct LLMs and three prevalent MAS. Compared to existing methods, the results demonstrate that M-Memory achieves statistically significant improvements in task success rate and generalization.

Our main contributions are summarized as follows: First, we propose M-Memory, a multi-perspective, hierarchical memory framework for MAS, which fosters efficient inter-agent collaboration by introducing multi-dimensional memory perspectives at various levels. Second, extensive experiments on three benchmarks demonstrate that M-Memory offers two primary advantages: (I) Superior Performance and (II) Resource Efficiency.

2 Preliminary

2.1 Multi-Agent System

A Multi-Agent System (MAS) consists of N agents collaborating through structured communication. Each agent X_i is equipped with an LLM backbone, a role specification Persona_i, a memory state Mem_i, and optional tools. Upon receiving query q, agents interact through T communication rounds, producing a final response $a^{(T)}$. Detailed agent interaction mechanics follow standard MAS frameworks (Zhuge et al., 2024; Wu et al., 2024; Qian et al., 2024b; Liu et al., 2023). The full Related Works are in Appendix D.

2.2 Memory Architecture

M-Memory organizes memory into three interconnected structures centered around a pivotal *Query Graph*. This architecture implements a bidirectional knowledge flow: fine-grained agent interactions during task execution are captured downward to construct the *Interaction Trajectory*, while execution outcomes are abstracted upward to distill high-level cognitive insights for a *Insight Graph*. By anchoring low-level behavioral traces and strategic summaries to the central query repository, this tripartite design bridges experiential records with actionable knowledge.

Interaction Trajectory. For each task q, the interaction trajectory is stored as $\mathcal{T}_{\text{inter}}^{(q)}$, capturing agent utterances and their dependencies during task execution.

Query Graph. The Query Graph $\mathcal{G}_{\text{query}} = (\mathcal{Q}, \mathcal{E}_q)$ stores historical queries and their relationships:

$$Q_i = (q_i, \Psi_i, \mathcal{T}_{\text{inter}}^{(q_i)}) \in \mathcal{Q}, \tag{1}$$

where the query node Q_i consists of a task description q_i, a task execution status $\Psi_i \in$

$\{\text{Solved}, \text{Failed}\}$, and the interaction trajectory $\mathcal{T}_{\text{inter}}^{(q_i)}$. Edges $\mathcal{E}_q$ encode semantic similarity between queries. Details can refer to Section 3.3.

Insight Graph. The Insight Graph $\mathcal{G}_{\text{insight}} = (\mathcal{I}, \mathcal{E}_i)$ captures distilled knowledge from three cognitive perspectives:

$$\mathcal{I} = \mathcal{I}_S \cup \mathcal{I}_P \cup \mathcal{I}_C, \tag{2}$$

where $\mathcal{I}_S$ (**Shortcuts**) represents Reusable strategies from similar successful tasks, $\mathcal{I}_P$ (**Pitfalls**) represents Failure patterns from similar failed tasks, and $\mathcal{I}_C$ (**Cross-domain Consensus**) represents Generalizable principles from diverse successful tasks. Each insight node $\iota_i = (\kappa_i, \Omega_i)$ contains insight content κ_i and a set of queries $\Omega_i \subseteq \mathcal{Q}$ that are successfully executed based on κ_i. The edges $\mathcal{E}_i$ forming hyper-connections where (ι_i, ι_j, Q_k) indicates insight ι_i contextualizes ι_j through query Q_k. Specifically, when two insights correspond to the same query, they implicitly share a contextual linkage.

3 Method

This section details the M-Memory workflow (Figure 2). Upon receiving a new task q, M-Memory performs: (1) coarse-grained retrieval for relevant queries, (2) bidirectional hierarchical traversal for multi-perspective memory, and (3) synchronous memory update based on execution feedback.

3.1 Coarse-grained Memory Retrieval

When encountering a task q, M-Memory first retrieves the top-k most similar tasks $\mathcal{Q}^S$ from Query Graph $\mathcal{G}_{\text{query}}$ with cosine similarity:

$$\text{sim}(q, q_i) = \cos\big(\mathbf{v}(q), \mathbf{v}(q_i)\big), \tag{3}$$

where $\mathbf{v}(\cdot)$ maps tasks to embeddings.

To mitigate noise from pure semantic similarity, we expand $\mathcal{Q}^S$ via 2-hop expansion:

$$\tilde{\mathcal{Q}}^S = \mathcal{Q}^S \cup \bigcup_{Q_i \in \mathcal{Q}^S} \mathcal{N}^2(Q_i), \tag{4}$$

where $\mathcal{N}^2(Q_i) = \{Q_j \mid \text{dist}_{\mathcal{G}_{\text{query}}}(Q_i, Q_j) \leq 2\}$.

Subsequently, to provide the current task q with memory support from more diverse task scenarios—while ensuring the retrieved task nodes have sufficient relevance to offer positive assistance—M-Memory performs a retrieval on the query graph $\mathcal{G}_{\text{query}}$ based on the Maximal Marginal Relevance (MMR) score (Carbonell and Goldstein, 1998). From the top-k results, this process produces a diverse query set $\mathcal{Q}^D$ where $\lambda = 0.5$ is a fixed trade-off parameter:

$$\begin{aligned}\text{MMR}(q, q_i) =& \lambda \cdot \text{sim}(q, q_i) \\ &- (1-\lambda) \cdot \max_{Q_j \in \mathcal{Q}^D} \text{sim}(q_i, q_j),\end{aligned} \tag{5}$$

followed by 2-hop expansion: $\tilde{\mathcal{Q}}^D = \mathcal{Q}^D \cup \bigcup_{Q_i \in \mathcal{Q}^D} \mathcal{N}^2(Q_i)$.

3.2 Bi-directional Memory Traversal

Given $\tilde{\mathcal{Q}}^S$ and $\tilde{\mathcal{Q}}^D$, M-Memory traverses memory bidirectionally.

Upward Traversal (High-Level Insights). M-Memory first performs an upward traversal to retrieve insight sets $\mathcal{S}^S$, $\mathcal{S}^P$, and $\mathcal{S}^C$ from multiple perspectives that may offer high-level cognitive guidance for the current task:

$$\begin{aligned}\mathcal{S}^S &= \{\iota_i^s \in \mathcal{I}_S \mid \Omega_i^s \cap \tilde{\mathcal{Q}}^S \neq \emptyset\}, \\ \mathcal{S}^P &= \{\iota_i^p \in \mathcal{I}_P \mid \Omega_i^p \cap \tilde{\mathcal{Q}}^S \neq \emptyset\}, \\ \mathcal{S}^C &= \{\iota_i^c \in \mathcal{I}_C \mid \Omega_i^c \cap \tilde{\mathcal{Q}}^D \neq \emptyset\}.\end{aligned} \tag{6}$$

Downward Traversal (Low-Level Trajectories). In addition to generalized, multi-perspective insights, the fine-grained textual interaction histories within the MAS are also of significant value, as they reveal the underlying reasoning patterns that lead to successful or failed collaborations. To utilize such information more effectively, M-Memory performs a downward traversal employing an LLM-based graph sparsifier, $\mathcal{S}_{\text{LLM}}(\cdot)$, to extract core trajectory that contain key inter-agent cooperation (detailed prompts are in Appendix C.1):

$$\begin{aligned}q_s^* &= \text{argmax}_{Q_i \in \tilde{\mathcal{Q}}^S} \text{Rel}(q, q_i, 1), \\ q_p^* &= \text{argmax}_{Q_i \in \tilde{\mathcal{Q}}^S} \text{Rel}(q, q_i, 0), \\ q_c^* &= \text{argmax}_{Q_i \in \tilde{\mathcal{Q}}^D} \text{MMR}(q, q_i), \\ \tilde{\mathcal{T}}_{\text{inter}} &= \mathcal{S}_{\text{LLM}}\big(\mathcal{T}_{\text{inter}}^{(q_s^*)}, \mathcal{T}_{\text{inter}}^{(q_p^*)}, \mathcal{T}_{\text{inter}}^{(q_c^*)}, q\big),\end{aligned} \tag{7}$$

where $\text{Rel}(q, q_i, x)$ (Appendix C.2) evaluates task relevance via LLMs. If $x = 1$, the historical task is designated as successful; conversely, it is marked as failed. By identifying and preserving essential dialogue elements, the sparsifier $\mathcal{S}_{\text{LLM}}(\,\cdot\,)$ synthesizes and summarizes the raw interactions from multi-perspective tasks. This process extracts $\tilde{\mathcal{T}}_{\text{inter}}$ that provide comprehensive low-level guidance for executing task q.

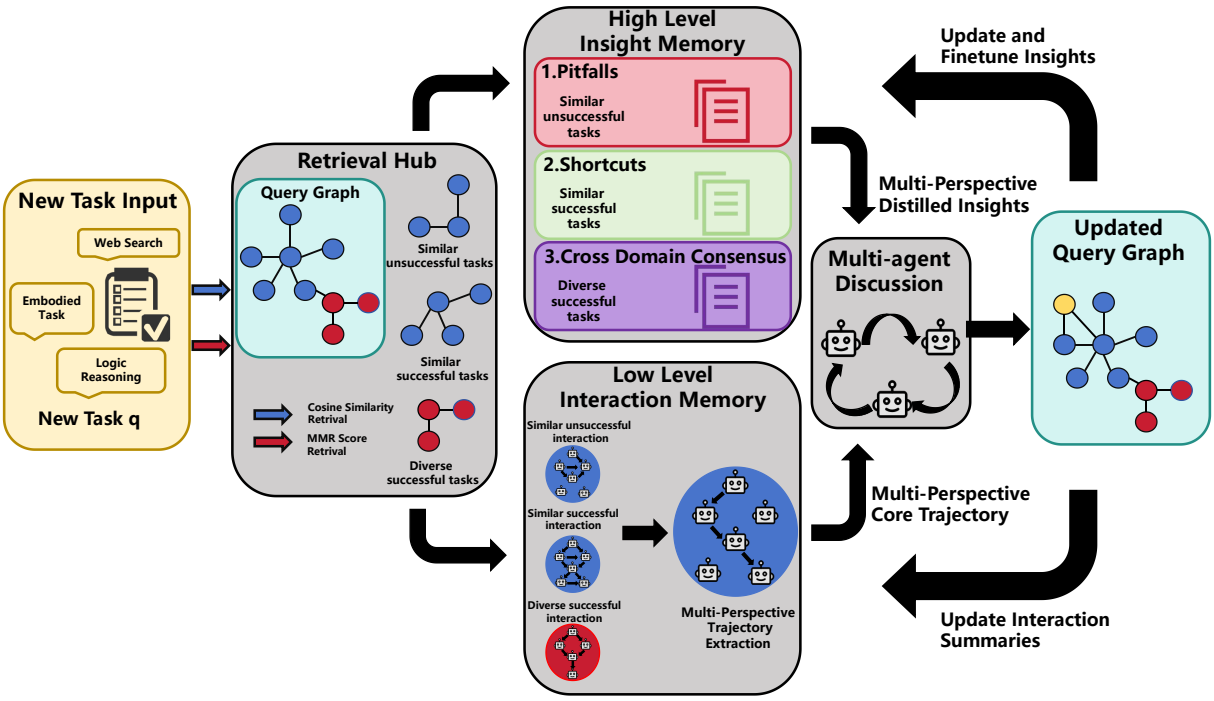

Figure 2: The M-Memory framework. A new query activates retrieval on the Query Graph, leading to a bidirectional traversal: an upward pass yields High-Level Insights (Pitfalls, Shortcuts, Consensus), while a downward pass extracts a Core Trajectory from Low-Level Interaction Memory. Both outputs jointly guide the Multi-agent Discussion. Post-execution feedback then drives a closed-loop update across the entire memory hierarchy, facilitating continuous self-evolution.

Agent Memory Initialization. Upon completion of the bidirectional traversal, we have obtained multi-perspective insights and a set of more refined, generalizable collaborative trajectories. Subsequently, M-Memory provides specialized memory support for each agent X_i in the MAS to initialize their internal memory state Mem_i (Appendix C.4):

$$\text{Mem}_i \leftarrow \Phi\big(\mathcal{S}^S, \mathcal{S}^P, \mathcal{S}^C, \tilde{\mathcal{T}}_{\text{inter}}; \text{Persona}_i, q\big), \quad (8)$$

where the operator $\Phi(\cdot)$ evaluates the utility and relevance of each insight $\iota_i \in \mathcal{S}^S \cup \mathcal{S}^P \cup \mathcal{S}^C$ and sparsified interaction trajectory $\tilde{\mathcal{T}}_{\text{inter}}$ to the agent-specific role Persona_i and task q.

3.3 Hierarchical Memory Update

After task completion with status Ψ, M-Memory updates its hierarchical memory architecture to incorporate this new task.

Interaction-Level. At the interaction level, M-Memory tracks each agent's utterances during the task execution to construct the interaction trajectory $\mathcal{T}_{\text{inter}}^{(q)}$ and stores the trajectory.

Query-Level. At the query level, a new query node is instantiated and added to the query graph:

$$Q_{\text{new}} = (q, \Psi, \mathcal{T}_{\text{inter}}^{(q)}). \quad (9)$$

New edges $\mathcal{E}_{\text{new}}$ connect Q_{new} to related historical queries, with each edge attribute capturing the pairwise cosine similarity of their task descriptions:

$$\mathcal{E}_{\text{new}} = \{(Q, Q_{\text{new}}) \mid Q \in \text{ConnSet}(\Psi)\}, \quad (10)$$

$$\text{ConnSet}(\Psi) = \begin{cases} \mathcal{Q}_M \cup \{Q \mid \iota_i \in \mathcal{S}^M, Q \in \Omega_i\} \\ \text{if } \Psi = \text{Solved}, \\ \mathcal{Q}_M \\ \text{if } \Psi = \text{Failed}, \end{cases} \quad (11)$$

where $\text{ConnSet}(\Psi)$ represents the set of nodes that will be connected to the new node Q_{new}, $\mathcal{S}^M = \mathcal{S}^S \cup \mathcal{S}^P \cup \mathcal{S}^C$, and $\mathcal{Q}_M \subseteq \mathcal{Q}^S$ denotes the top-m similar queries. The value of m is set to 10.

Insight-Level. At this level, M-Memory incrementally updates $\mathcal{G}_{\text{insight}}$ by distilling cognitive patterns from the completed task q. We first identify the subset of retrieved insights that effectively guided the multi-agent reasoning, denoted as $\mathcal{I}_{\text{ret}} \subseteq \mathcal{S}^M$. For each validated insight $\iota_i \in \mathcal{I}_{\text{ret}}$, if the execution status of the new node is successful, we reinforce its applicability by expanding its support set in $\mathcal{I}$ (2) with the new query node Q_{new}:

$$\Omega_i = \Omega_i \cup \{Q_{\text{new}}\}, \quad \forall \iota_i \in \mathcal{I}_{\text{ret}}. \quad (12)$$

Otherwise, no update is performed.

Concurrently, we extract novel patterns by summarizing the interaction trajectory $\mathcal{T}_{\text{inter}}^{(q)}$ and execution status Ψ via function $\mathcal{J}(\cdot)$ (Appendix C.5) to construct a new insight node:

$$\iota_{\text{new}} = (\mathcal{J}(Q_{\text{new}})), \{Q_{\text{new}}\}). \tag{13}$$

which is then added into the updated $\mathcal{I}$ (2):

$$\mathcal{I} = \mathcal{I} \cup \{\iota_{\text{new}}\}. \tag{14}$$

The hyper-edges connecting ι_{new} to its guiding insights are established as:

$$\mathcal{E}_{i,\text{new}} = \left\{(\iota_i, \iota_{\text{new}}, Q_{\text{new}}) \mid \iota_i \in \mathcal{I}_{\text{ret}}\right\}. \tag{15}$$

Subsequently, the corresponding insights in $\mathcal{I}$ (2), defined by Eq. (6), are updated to align with actual task feedback. According to the execution status Ψ, successful executions trigger the abstraction of shortcuts and consensus via the fine-tuning operator $\mathcal{F}_{\text{LLM}}$ (Appendix C.3):

$$\mathcal{S}^S = \mathcal{F}_{\text{LLM}}(\mathcal{S}^S, q), \quad \mathcal{S}^C = \mathcal{F}_{\text{LLM}}(\mathcal{S}^C, q). \tag{16}$$

while failures prompt updates to pitfall patterns:

$$\mathcal{S}^P = \mathcal{F}_{\text{LLM}}(\mathcal{S}^P, q). \tag{17}$$

Finally, with the refined insights and newly consolidated edges, the overall insight graph $\mathcal{G}_{\text{insight}}$ are updated for subsequent task execution.

4 Experiment

4.1 Experiment Setup

Datasets and Benchmarks. To ensure a rigorous and broadly applicable evaluation, we adhere to four benchmarks that have been extensively validated and repeatedly employed in prior studies across distinct domains: (1) Knowledge Reasoning: FEVER (Thorne et al., 2018) and HotpotQA(Yang et al., 2018); (2) Embodied Action: ALFWorld (Shridhar et al., 2020); and (3) Gaming: PDDL (Chang et al., 2024). These datasets are widely recognized for their reliability, standardized annotation protocols, and proven ability to stress-test memory-dependent capabilities. Detailed specifications are provided in Appendix A.1. **Baselines.** For comparative analysis, we select four widely adopted single-agent baselines: a standard non-memory model, Voyager (Wang et al., 2023), MemoryBank (Zhong et al., 2024), and Generative Agents (Park et al., 2023). Furthermore, we incorporate three extensively utilized multi-agent memory architectures from MetaGPT (Hong et al., 2024), ChatDev (Qian et al., 2024a), and G-Memory (Zhang et al., 2025a) (Details are in Appendix A.2). These frameworks have been thoroughly validated across multiple independent studies and serve as highly reliable reference points for evaluating memory-augmented agent performance. **MAS and LLM Backbones.** To guarantee robust and reproducible multi-agent execution, we select three representative multi-agent frameworks for integration with M-Memory and the baselines: AutoGen (Wu et al., 2024), DyLAN (Liu et al., 2023), and MacNet (Qian et al., 2024b) (Details are in Appendix A.3). For the instantiation of these MAS frameworks, we employ an open-source LLM, Qwen-2.5-14b, as well as a proprietary LLM, gpt-4o-mini. The Qwen series is deployed via local instantiation with vllm, while the GPT models are invoked through the OpenAI API. **Parameter Settings.** We adopt ALL-MINILM-L6-V2 (Wang et al., 2020) as the embedding function $v(\cdot)$ in Eq. (3). The number of k in Section 3.1 is selected as 1. The number of hop expansion in Eq. (4) is selected as 2. To rigorously assess the long-term self-evolution capacity of memory architectures, inspired by Reflexion (Shinn et al., 2023a), all methods undergo three consecutive evaluation runs on identical datasets, and the average result of the three runs was taken as the final result. Crucially, each method's memory state is retained across runs, enabling a reliable assessment of cumulative learning and the theoretical upper bound of performance.

4.2 Main Results

Tables 1 and 3 comprehensively report the performance of various memory architectures across two LLM backbones and three MAS frameworks (detailed evaluation metrics are in Appendix A.1).

As shown in Table 1, when integrated with frameworks driven by gpt-4o-mini, M-Memory yields significant gains. Compared to the best-performing memory baselines, M-Memory achieves an average performance increase of: 3.19% with Autogen, 7.69% with Dylan, and 6.94% with MacNet. These results indicate that even with a more compact model like gpt-4o-mini, M-Memory's structured approach to recalling and summarizing past interactions provides a substantial advantage. In addition, the performance uplift from M-Memory becomes even more pronounced

Table 1: Performance comparison with different memory architectures on three benchmarks. The underlying LLM backbone is gpt-4o-mini. Compared to the best-performing baselines, M-Memory achieves an average performance increase of: 3.19% with Autogen, 7.69% with Dylan, and 6.94% with MacNet. (Qwen results see Appendix B)

MAS	Method	Venue	Alfworld	PDDL	PDDL(RE)	FEVER	HotpotQA	Avg.
Autogen	No-memory	–	70.74	8.33	22.4	61.67	51.33	42.89
	Voyager	NeurIPS'23	77.31	6.67	27.08	61.33	44.67	43.41
	MemoryBank	AAAI'24	72.39	1.67	11.95	62.33	47.33	39.13
	Generative	UIST'23	85.07	6.67	25.59	63.33	54.33	47.00
	MetaGPT	ICLR'24	82.09	8.33	31.19	61.67	54.33	47.52
	ChatDev	ACL'24	70.15	5.00	18.63	59.67	53.67	41.42
	G-Memory	NeurIPS'25	83.58 [1]	16.67	37.04	66.33	57.33	52.19
	M-Memory	proposed	**86.57**	**16.67**	**38.69**	**66.33**	**68.33**	**55.32**
Dylan	No-memory	–	73.13	1.67	9.30	61.33	51.67	39.42
	Voyager	NeurIPS'23	75.37	1.67	9.00	63.33	44.33	38.74
	MemoryBank	AAAI'24	61.19	1.67	19.29	61.00	47.67	38.16
	Generative	UIST'23	82.08	3.33	11.89	66.67	54.33	43.66
	MetaGPT	ICLR'24	80.60	8.33	31.06	60.67	54.67	47.07
	ChatDev	ACL'24	61.94	1.67	5.14	62.67	56.33	37.55
	G-Memory	NeurIPS'25	79.10	10.00	25.03	65.67	56.33	47.23
	M-Memory	proposed	**84.70**	**11.67**	**39.58**	**74.67**	**64.00**	**54.92**
MacNet	No-memory	–	63.43	6.67	24.15	64.33	48.33	41.38
	Voyager	NeurIPS'23	70.90	13.33	31.57	62.67	47.67	45.23
	MemoryBank	AAAI'24	55.97	10.00	32.22	56.00	50.00	40.84
	Generative	UIST'23	73.13	10.00	33.73	57.33	52.33	45.30
	MetaGPT	ICLR'24	77.61	11.67	36.78	65.67	54.00	49.15
	ChatDev	ACL'24	62.16	1.67	18.80	64.00	51.67	39.66
	G-Memory	NeurIPS'25	72.38	5.00	20.56	63.67	58.33	43.99
	M-Memory	proposed	**84.33**	**20.00**	**38.81**	**72.00**	**65.33**	**56.09**

when paired with the more powerful Qwen-2.5-14b backbone, as detailed in Table 3 (Appendix B). In this configuration, M-Memory's average improvement over the strongest baselines surges to: 7.89% with Autogen, 8.97% with Dylan, and 7.61% with MacNet. This highlights a clear synergistic effect: a more capable LLM backbone can better leverage the rich, structured, and diverse memories curated by M-Memory. Notably, other baselines struggled significantly on the PDDL dataset due to the requirement for robust causal reasoning and experiential transfer capabilities. In contrast, M-Memory significantly enhances the performance of multi-agent systems on the PDDL(RE) benchmark. Specifically, within the Dylan system utilizing the Qwen-2.5-14B backbone, performance increased from 9.3% to 39.58%, yielding an improvement of 30.28%. These results demonstrate that, driven by its core mechanisms, M-Memory excels in such complex scenarios: **Learning from Failure**: By retrieving the most similar failed tasks, M-Memory provides crucial negative guidance, assisting agents in avoiding common pitfalls and dead ends inherent in long-horizon tasks. **Strategy Diversity**: The retrieval of diverse successful tasks based on MMR provides a rich repertoire of "generalizable" strategies, preventing agents from becoming trapped in single, sub-optimal approaches.

In summary, the empirical results provide strong and multifaceted evidence for our central hypothesis. The consistent gains across different LLMs and agent frameworks confirm the general applicability and robustness of M-Memory.

4.3 Framework Analysis

Ablation Study. To systematically evaluate the contribution of M-Memory's architectural design, we conduct a comprehensive ablation study with the Qwen-2.5-14B backbone (Table 2). We assess performance degradation under two experimental axes: (i) incremental removal of cognitive perspectives, and (ii) token-budget constrained comparison against G-Memory.

Impact of Cognitive Perspectives. We map *One Perspective* to the retrieval of successful task summaries ('shortcut'), and *Two Perspective* to the joint incorporation of both successful and failed task summaries ('pitfall'). Restricting the memory to a single perspective significantly impairs performance, degrading the average scores across AutoGen, DyLAN, and MacNet by 7.89%, 8.97%, and 7.61%, respectively, relative to the full three-perspective configuration. Even when both success-

[1] We failed to replicate the performance of the GMemory method (in Autogen with gpt-4o-mini) after multiple attempts using the provided open-source code.

Table 2: Ablation study of M-Memory. The performance trend confirms that integrating a comprehensive set of cognitive perspectives is critical for maximizing task execution quality. The LLM backbone is Qwen-2.5-14b.

MAS	Memory	Alfworld	PDDL	PDDL(RE)	FEVER	HotpotQA	Avg.
AutoGen	G-Memory$_{\text{same}}$	86.19	35.00	53.89	67.67	61.67	60.88
	One Perspective	85.08	33.34	55.43	70.33	62.33	61.30
	Two Perspective	85.82	39.17	59.93	74.67	62.67	64.45
	Three Perspective	91.04	43.33	64.22	78.67	68.67	69.19
DyLAN	G-Memory$_{\text{same}}$	85.07	31.67	54.89	67.33	56.33	59.06
	One Perspective	83.58	30.00	51.23	66.33	55.33	57.29
	Two Perspective	84.33	35.84	60.08	69.33	55.67	61.05
	Three Perspective	87.31	41.67	66.96	75.67	59.67	66.26
MacNet	G-Memory$_{\text{same}}$	80.97	26.67	47.34	63.33	51.33	53.93
	One Perspective	77.61	26.67	48.73	69.72	50.67	54.68
	Two Perspective	82.84	31.67	54.01	72.33	61.33	60.44
	Three Perspective	84.33	35.00	55.43	73.67	63.00	62.29

ful and failed insights are utilized, performance remains suboptimal, exhibiting average drops of 4.74%, 5.21%, and 1.85%. This performance trend confirms that integrating a comprehensive set of cognitive perspectives is critical for maximizing task execution quality.

Context Budget Analysis. To isolate the influence of the memory architecture from context capacity, we quantified the token consumption of M-Memory and constrained G-Memory's token budget to be no less than that of M-Memory by introducing a larger number of insights into G-Memory. The results for G-Memory$_{\text{same}}$ indicate that merely expanding the context window yields marginal gains, underscoring that M-Memory's performance advantage stems from its structured cognitive framework rather than raw token volume. Collectively, these findings validate the necessity of M-Memory's multi-dimensional design, confirming that synergistic perspective integration enhances both task performance and self-evolutionary capacity.

Cost Analysis. To evaluate the token efficiency of M-Memory, we illustrate the trade-off between task performance and token cost for different memory architectures in Fig. 3. Our findings indicate that: M-Memory achieves higher-performance collective memory while maintaining a stable and controlled token cost. As shown in Fig. 3, M-Memory delivers the highest performance improvement on the PDDL(RE)+AutoGen and PDDL+AutoGen datasets (19.05% and 18.33% over the no-memory setting, respectively), with an increase in token cost of only 5.0×10^6. In contrast, GMemory yields smaller gains of 11.91% and 10% yet incurs a greater additional cost of 7.1×10^6. This clearly demonstrates the superior token efficiency of M-Memory.

Sensitivity Analysis. For hop expansion, as depicted in Fig. 4 (a), a 2-hop expansion consistently achieves the best or near-best performance across all tasks, reaching peak accuracies under the AutoGen framework of 91.04% (Alfworld), 64.22% (PDDL(RE)), 43.33% (PDDL), and 78.67% (FEVER). In contrast, 1-hop and 3-hop settings often lead to performance degradation; for instance, on PDDL(RE), performance drops to 57.92% with a 3-hop expansion. This suggests that an appropriate hop expansion can enrich the architecture's memory perspective, whereas an excessive expansion may introduce irrelevant information, thereby impairing the agent's reasoning capabilities. Furthermore, Fig. 4 (b) shows that the optimal value for k is 1. Within the AutoGen and DyLAN frameworks, while increasing k can potentially introduce more diverse perspectives and improve system performance, it incurs a substantially higher token cost. This indicates a low cost-benefit ratio for retrieving additional queries. Consequently, we adopt a 2-hop expansion and $k = 1$ throughout our experiments.

4.4 Case Study

Figure 5 illustrates how M-Memory synthesizes and delivers multi-faceted memory cues to guide a complex block-manipulation task within the PDDL environment. Upon receiving the initial task, M-Memory retrieves several related queries and decomposes the stored knowledge into four complementary perspectives to steer the agent's decision-making. At the macro level, the **Cross-domain Consensus** insight provides a high-level assembly strategy, explicitly advising the agent to prioritize

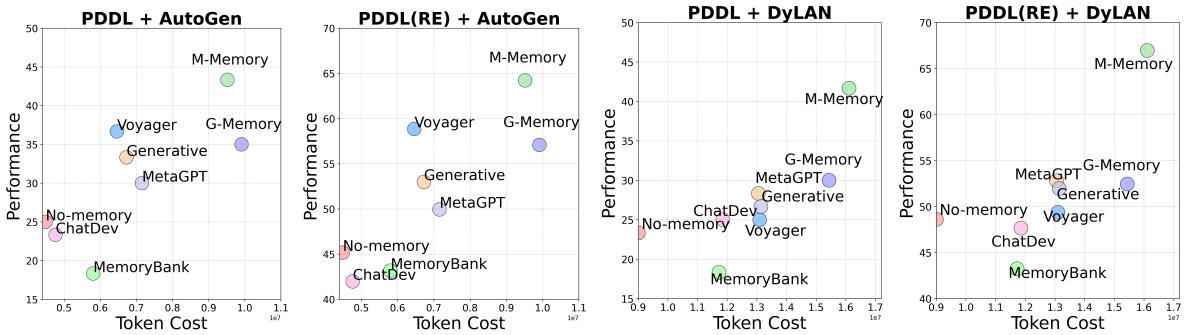

Figure 3: The relationship between performance and total system token cost across different memory architectures. M-Memory achieves higher-performance collective memory while maintaining a stable and controlled token cost.

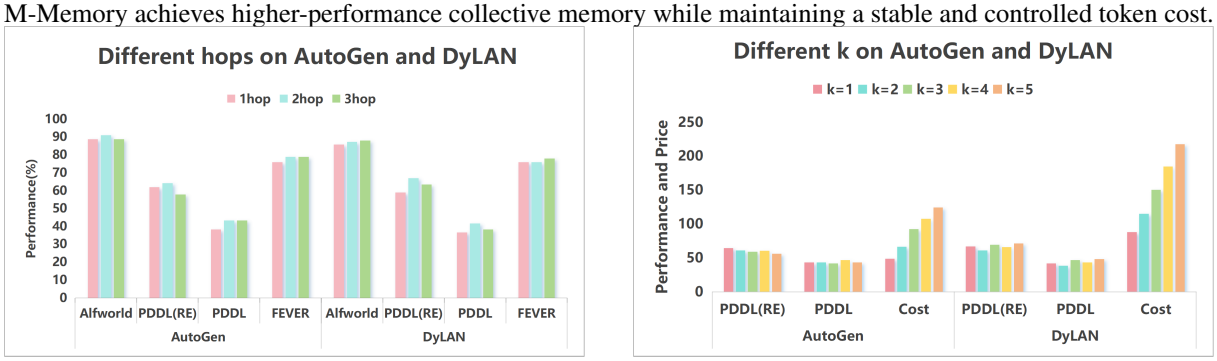

Figure 4: (a) Sensitivity analysis of the hop expansion; (b) Sensitivity analysis of the number of selected queries k.

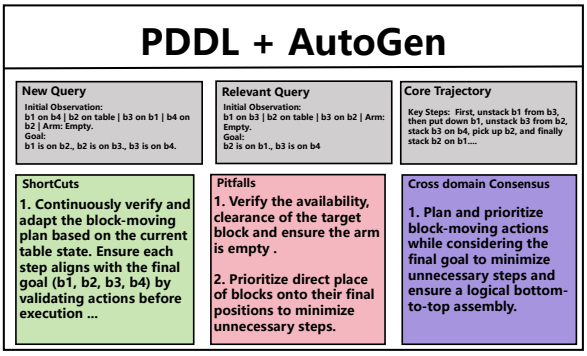

Figure 5: Case study of M-Memory.

moves with the final tower configuration in mind and enforce a logical bottom-to-top sequence to systematically eliminate redundant intermediate steps. For dynamic execution, the **ShortCuts** insight offers real-time verification protocols, instructing the system to continuously adapt the manipulation plan based on the instantaneous table state and validate each primitive action against the terminal goal to prevent irreversible dead-end states. Concurrently, the **Pitfalls** insights inject critical operational constraints, warning the agent to verify target clearance and gripper availability before grasping while advocating direct placement to bypass unnecessary pick-and-place cycles. Finally, M-Memory surfaces a **Core Trajectory** that translates these abstract guidelines into concrete primitives, outlining a precise step-by-step sequence (*e.g.*, unstacking $b1$ from $b3$, relocating $b3$ to $b4$, and stacking $b2$ onto $b1$) to directly satisfy the target conditions. By integrating strategic foresight, adaptive validation, constraint-based risk mitigation, and explicit execution roadmaps, M-Memory transforms fragmented historical experiences into a cohesive, multi-level decision-support framework that significantly enhances task reliability and execution efficiency.

5 Conclusion

To overcome the limited self-evolution of existing multi-agent systems (MAS) caused by their singular cognitive perspective, we propose M-Memory, a hierarchical framework that reshapes both high-level cognitive and low-level interaction memories from three complementary perspectives. At the high level, M-Memory distills shortcut insights from similar successful tasks, pitfall insights from similar failed tasks, and cross-domain consensus insights from dissimilar successful tasks. At the low level, it enriches the summarization of key steps with trajectories from both similar failures and dissimilar successes, thereby improving robustness and generalization. Extensive experiments show that M-Memory can be seamlessly integrated into state-of-the-art MAS frameworks and substantially enhances their self-evolutionary capabilities.

Limitations

While M-Memory substantially improves system performance using its multi-dimensional cognitive perspectives, the opportune timing for introducing each perspective remains a critical area for future work. Appropriate timing could enable the system to evolve its capabilities on specific tasks to their performance ceiling more rapidly, thereby substantially reducing costs.

References

J. Carbonell and J. Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the ACM SIGIR*, pages 335–336.

M. Chang, J. Zhang, Z. Zhu, C. Yang, Y. Yang, Y. Jin, Z. Lan, L. Kong, and J. He. 2024. Agentboard: An analytical evaluation board of multi-turn llm agents. *Advances in neural information processing systems*, 37:74325–74362.

W. Chen, Y. Su, J. Zuo, C. Yang, C. Yuan, C. Chan, H. Yu, Y. Lu, Y. Hung, C. Qian, and 1 others. 2023. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *Proceedings of the ICLR*.

P. Chhikara, D. Khant, S. Aryan, T. Singh, and D. Yadav. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.

D. Driess, F. Xia, M. SM Sajjadi, C. Lynch, A. Chowdhery, A. Wahid, J. Tompson, Q. Vuong, T. Yu, We. Huang, and 1 others. 2023. Palm-e: an embodied multimodal language model. In *Proceedings of the ICML*.

Y. Du, S. Li, A. Torralba, J. B Tenenbaum, and I. Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate, 2023. *URL https://arxiv. org/abs/2305.14325*, 3.

L. E. Erdogan, N. Lee, S. Kim, S. Moon, H. Furuta, G. Anumanchipalli, K. Keutzer, and A. Gholami. 2025. Plan-and-act: Improving planning of agents for long-horizon tasks. *arXiv preprint arXiv:2503.09572*.

C. Gao, X. Lan, N. Li, Y. Yuan, J. Ding, Z. Zhou, F. Xu, and Y. Li. 2024. Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Humanities and Social Sciences Communications*, 11(1):1–24.

S. Hong, X. Zheng, J. Chen, Y. Cheng, J. Wang, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou, and 1 others. 2024. Metagpt: Meta programming for multi-agent collaborative framework. In *Proceedings of the ICLR*, volume 10.

C. Hu, J. Fu, C. Du, S. Luo, J. Zhao, and H. Zhao. 2023. Chatdb: Augmenting llms with databases as their symbolic memory. *arXiv preprint arXiv:2306.03901*.

S. Hu, C. Lu, and J. Clune. 2024a. Automated design of agentic systems. *arXiv preprint arXiv:2408.08435*.

Y. Hu, Y. Cai, Y. Du, X. Zhu, X. Liu, Z. Yu, Y. Hou, S. Tang, and S. Chen. 2024b. Self-evolving multi-agent collaboration networks for software development. *arXiv preprint arXiv:2410.16946*.

G. Li, H. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem. 2023. Camel: Communicative agents for" mind" exploration of large language model society. *Proceedings of the NIPS*, 36:51991–52008.

X. Li, S. Wang, S. Zeng, Y. Wu, and Y. Yang. 2024. A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9.

X. Liang, Y. He, Y. Xia, X. Song, J. Wang, M. Tao, L. Sun, X. Yuan, J. Su, K. Li, and 1 others. 2024. Self-evolving agents with reflective and memory-augmented abilities. *arXiv preprint arXiv:2409.00872*.

Z. Liu, Y. Zhang, P. Li, Y. Liu, and D. Yang. 2023. Dynamic llm-agent network: An llm-agent collaboration framework with agent team optimization. *arXiv preprint arXiv:2310.02170*.

J. Lu, S. An, M. Lin, G. Pergola, Y. He, D. Yin, X. Sun, and Y. Wu. 2023. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239*.

T. Masterman, S. Besen, M. Sawtell, and A. Chao. 2024. The landscape of emerging ai agent architectures for reasoning, planning, and tool calling: A survey. *arXiv preprint arXiv:2404.11584*.

Y. Ou, W. Zhou, S. Ding, L. Li, J. Wu, T. Wang, J. Chen, S. Wang, X. Xu, N. Zhang, and 1 others. 2025. Symbolic learning enables self-evolving agents. *AI Open*.

C. Packer, S. Wooders, K. Lin, V. Fang, S. G. Patil, I. Stoica, and J. E. Gonzalez. 2023. MemGPT: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*.

J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the UIST*, pages 1–22.

G. Piatti, Z. Jin, M. Kleiman-Weiner, B. Schölkopf, M. Sachan, and R. Mihalcea. 2024. Cooperate or collapse: Emergence of sustainability behaviors in a society of llm agents. *CoRR*.

R. Pina, V. De Silva, and C. Artaud. 2025. Discovering causality for efficient cooperation in multi-agent environments. *Neurocomputing*, page 130358.

Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. 2024. Agent q: Advanced reasoning and learning for autonomous ai agents. *arXiv preprint arXiv:2408.07199*.

C. Qian, X. Cong, C. Yang, W. Chen, Y. Su, J. Xu, Z. Liu, and M. Sun. 2024a. Communicative agents for software development. In *Proceedings of the ACL*, volume 6, page 1.

C. Qian, Z. Xie, Y. Wang, W. Liu, K. Zhu, H. Xia, Y. Dang, Z. Du, W. Chen, C. Yang, and 1 others. 2024b. Scaling large language model-based multi-agent collaboration. *arXiv preprint arXiv:2406.07155*.

R. Salama, J. Cai, M. Yuan, A. Currey, M. Sunkara, Y. Zhang, and Y. Benajiba. 2025. Meminsight: Autonomous memory augmentation for llm agents. *arXiv preprint arXiv:2503.21760*.

N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao. 2023a. Reflexion: Language agents with verbal reinforcement learning. *Advances in NIPS*, 36:8634–8652.

N. Shinn, B. Labash, and A. Gopinath. 2023b. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*.

M. Shridhar, X. Yuan, M. Côté, Y. Bisk, A. Trischler, and M. Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

P. Singh. 2003. Examining the society of mind. *Computing and Informatics*, 22(6):521–543.

X. Tang, T. Hu, M. Ye, Y. Shao, X. Yin, S. Ouyang, W. Zhou, P. Lu, Z. Zhang, Y. Zhao, and 1 others. 2025. Chemagent: Self-updating library in large language models improves chemical reasoning. *arXiv preprint arXiv:2501.06590*.

J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. In *Proceedings of the NIPS*.

J. Wang, J. Wang, B. Athiwaratkun, C. Zhang, and J. Zou. 2024a. Mixture-of-agents enhances large language model capabilities. *arXiv preprint arXiv:2406.04692*.

L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, and 1 others. 2024b. A survey on large language model based autonomous agents. *Front. Comput. Sci.*, 18(6):186345.

S. Wang, Z. Yu, X. Jiang, S. Lan, M. Shi, N. Chang, J. Kautz, Y. Li, and J. M Alvarez. 2024c. Omnidrive: A holistic llm-agent framework for autonomous driving with 3d perception, reasoning and planning. *CoRR*.

W. Wang, D. Zhang, T. Feng, B. Wang, and J. Tang. 2024d. Battleagentbench: A benchmark for evaluating cooperation and competition capabilities of language models in multi-agent systems. *arXiv preprint arXiv:2408.15971*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788.

Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, and 1 others. 2024. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *Proceedings of the COLM*.

Z. Xi, W. Chen, X. Guo, Wei He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou, and 1 others. 2025. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101.

W. Xu, Z. Liang, K. Mei, H. Gao, J. Tan, and Y. Zhang. 2025. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*.

Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the EMNLP*, pages 2369–2380.

Z. Yin, Q. Sun, C. Chang, Q. Guo, J. Dai, X. Huang, and X. Qiu. 2023. Exchange-of-thought: Enhancing large language model capabilities through cross-model communication. In *Proceedings of the EMNLP*, pages 15135–15153.

S. Yuan, K. Song, J. Chen, X. Tan, D. Li, and D. Yang. 2025. Evoagent: Towards automatic multi-agent generation via evolutionary algorithms. In *Proceedings of the NAACL*, pages 6192–6217.

G. Zhang, M. Fu, G. Wan, M. Yu, K. Wang, and S. Yan. 2025a. G-memory: Tracing hierarchical memory for multi-agent systems. *arXiv preprint arXiv:2506.07398*.

G. Zhang, L. Niu, J. Fang, K. Wang, L. Bai, and X. Wang. 2025b. Multi-agent architecture search via agentic supernet. *arXiv preprint arXiv:2502.04180*.

G. Zhang, Y. Yue, X. Sun, G. Wan, M. Yu, J. Fang, K. Wang, T. Chen, and D. Cheng. 2024a. G-designer: Architecting multi-agent communication topologies via graph neural networks. *arXiv preprint arXiv:2410.11782*.

J. Zhang, J. Xiang, Z. Yu, F. Teng, X. Chen, J. Chen, M. Zhuge, X. Cheng, S. Hong, J. Wang, and 1 others. 2024b. Aflow: Automating agentic workflow generation. *arXiv preprint arXiv:2410.10762*.

A. Zhao, D. Huang, Q. Xu, M. Lin, Y. Liu, and G. Huang. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI*, volume 38, pages 19632–19642.

L. Zheng, R. Wang, X. Wang, and B. An. 2024. Synapse: Trajectory-as-exemplar prompting with memory for computer control, 2024. *URL https://arxiv. org/abs/2306.07863*.

W. Zhong, L. Guo, Q. Gao, H. Ye, and Y. Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI*, volume 38, pages 19724–19731.

X. Zhu, Y. Chen, H. Tian, C. Tao, W. Su, C. Yang, G. Huang, B. Li, L. Lu, X. Wang, and 1 others. 2023. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144*.

Y. Zhu, S. Qiao, Y. Ou, S. Deng, S. Lyu, Y. Shen, L. Liang, J. Gu, H. Chen, and N. Zhang. 2025. Knowagent: Knowledge-augmented planning for llm-based agents. In *Proceedings of the NAACL*, pages 3709–3732.

M. Zhuge, W. Wang, L. Kirsch, F. Faccio, D. Khizbullin, and J. Schmidhuber. 2024. Gptswarm: Language agents as optimizable graphs. In *Proceedings of the ICML*.

A Experimental Details

A.1 Dataset Descriptions

This section details the datasets used for our experiments:

FEVER (Thorne et al., 2018) is a highly knowledge-intensive dataset focused on fact verification. Agents must verify claims by using a web search API, making it a standard benchmark for evidence-based reasoning. (Available at https://fever.ai/dataset/fever.html, under the Creative Commons Attribution-ShareAlike license). Given the large scale of the dataset, we randomly sampled 100 tasks for evaluation, proportionally reflecting the macro-level task distribution.

HotpotQA (Yang et al., 2018) is a multi-hop question answering dataset that provides rigorous supervision over supporting facts. It is designed to evaluate an agent's capability to retrieve and integrate information, with particular emphasis on its ability to perform interpretable reasoning via web search tools. (Available at https://hotpotqa.github.io/; licensed under CC BY-SA 4.0). Given the large scale of the dataset, we randomly sampled 100 tasks for evaluation, proportionally reflecting the macro-level task distribution.

ALFWorld (Shridhar et al., 2020) is a text-based embodied environment centered on household tasks, where agents use natural language instructions to navigate and interact with objects. (Available at https://alfworld.github.io/, under the MIT license).

PDDL is a game dataset provided by AgentBoard (Chang et al., 2024), containing a variety of strategic games where agents must complete complex tasks using PDDL expressions. (Available at https://github.com/hkust-nlp/AgentBoard, under a custom license).

Evaluation Metrics. We employ Success Rate for FEVER, HotpotQA, and ALFWORLD; Success Rate and Reward Rate (RE) for PDDL.

A.2 Baselines

For comparative analysis, we benchmark against a rigorously curated set of baselines that represent established milestones in the literature. Specifically, we evaluate three widely adopted single-agent memory systems: *Voyager* (Wang et al., 2023), which pioneers lifelong skill acquisition and iterative code generation in open-ended environments; *MemoryBank* (Zhong et al., 2024), which implements a cognitively grounded long-

term memory module featuring dynamic summarization, reflection, and temporal retrieval; and *Generative Agents* (Park et al., 2023), which utilizes a continuous memory stream integrated with observation–reflection–planning cycles to drive autonomous behavior. Furthermore, we incorporate three extensively validated multi-agent architectures: *MetaGPT* (Hong et al., 2024), which orchestrates agent collaboration through standardized operating procedures and shared context management; *ChatDev* (Qian et al., 2024a), which simulates role-based conversational workflows within a virtual software company to automate complex pipelines; and *GMemory* (Zhang et al., 2025a), which employs structured memory representations to enable efficient cross-agent knowledge sharing and state-aware reasoning. Collectively, these frameworks have been thoroughly benchmarked across multiple independent studies and serve as mature, highly reliable reference points for evaluating memory-augmented agent performance.

A.3 Multi-agent System Setup

This section details the configurations of the three MAS frameworks we employed: AutoGen, DyLAN, and MacNet.

AutoGen (Wu et al., 2024) is a popular framework for coordinating multi-agent collaboration to solve complex problems. We adopted its A3 decision architecture, which comprises three core components: (1) a Solver agent that generates solutions, initially prompted with "You are an agent designed for problem-solving"; (2) a Benchmark agent that evaluates the Solver's output against predefined criteria to identify errors; and (3) an Executor agent that converts the validated solution into executable instructions. This modular design enables transparent, verifiable, and actionable multi-agent collaboration.

DyLAN (Liu et al., 2023) is a debate-style framework, similar to LLM-Debate, that introduces a more efficient agent early-stopping mechanism for multi-round interactions. The framework uses an agent selection algorithm based on an unsupervised metric called the "Agent Importance Score," which identifies the most valuable agents through a task-specific pre-evaluation. In our implementation, three agents participate in the debate, while a fourth ranking agent evaluates their relative importance.

MacNet (Qian et al., 2024b) is a notable study on decentralized and scalable multi-agent systems. Its core innovation lies in replacing the central agent with edge agents. These edge agents are invoked during the interaction process to generate executable instructions for subsequent agents based on the outputs of preceding ones. In our implementation, we adopted MacNet's random graph topology (a structure proven to be robust in various scenarios) and deployed five additional agents to collaborate with the edge agents.

B Experimental Result

Table 3 presents the additional experimental results using Qwen-2.5-14b as the LLM backbone. In this configuration, M-Memory's average improvement over the strongest baselines surges to: 7.89% with Autogen, 8.97% with Dylan, and 7.61% with MacNet. This highlights a clear synergistic effect: a more capable LLM backbone can better leverage the rich, structured, and diverse memories curated by M-Memory.

C Prompt Set

C.1 Graph Sparsifier

Graph Sparsifier

You are an agent skilled at extracting key points. Given a task and a successful execution trajectory, your job is to identify the critical steps needed to complete the task while filtering out less important steps. Furthermore, if the given failure trajectory and diverse successful trajectory have a positive impact on the summary of steps, please make proper use of them. If not, keep the original steps unchanged.
Note:
- Strictly follow the original trajectory; absolutely no steps that are not in the trajectory should be added.
- Even in a successful trajectory, there may be some incorrect steps. Pay attention to actions that correspond to "Nothing happens" observations, as these actions are likely incorrect. Refer to the given failure trajectory to filter out these actions.
- You need to ensure that each step is at the finest granularity.
- You should strictly follow the output format in the example.

Table 3: Performance comparison with different memory architectures on three benchmarks. The underlying LLM backbone is Qwen-2.5-14b. Compared to the best-performing baselines, M-Memory achieves an average performance increase of: 7.89% with Autogen, 8.97% with Dylan, and 7.61% with MacNet.

MAS	Method	Venue	Alfworld	PDDL	PDDL(RE)	FEVER	HotpotQA	Avg.
Autogen	No-memory	–	80.22	27.5	45.81	64.33	51.33	53.84
	Voyager	NeurIPS'23	80.83	31.67	55.55	64.33	49.67	56.41
	MemoryBank	AAAI'24	77.62	20.83	44.75	65.67	51.33	52.04
	Generative	UIST'23	83.58	31.67	50.81	63.33	49.00	55.68
	MetaGPT	ICLR'24	80.08	27.5	52.49	63.33	54.67	55.61
	ChatDev	ACL'24	76.87	20.83	40.47	64.67	57.33	52.03
	G-Memory	NeurIPS'25	85.08	33.34	55.43	70.33	62.33	61.30
	M-Memory	proposed	**91.04**	**43.33**	**64.22**	**78.67**	**68.67**	**69.19**
Dylan	No-memory	–	76.87	24.17	47.92	59.33	53.33	52.32
	Voyager	NeurIPS'23	82.47	25.85	49.83	65.33	53.67	55.43
	MemoryBank	AAAI'24	78.36	18.33	44.03	58.67	55.33	50.94
	Generative	UIST'23	80.00	25.83	50.23	60.00	49.33	53.08
	MetaGPT	ICLR'24	80.97	27.50	53.13	61.67	53.33	55.32
	ChatDev	ACL'24	75.00	25.00	51.23	60.33	53.67	53.04
	G-Memory	NeurIPS'25	83.58	30.00	51.23	66.33	55.33	57.29
	M-Memory	proposed	**87.31**	**41.67**	**66.96**	**75.67**	**59.67**	**66.26**
MacNet	No-memory	–	61.20	23.33	42.95	65.33	53.33	49.23
	Voyager	NeurIPS'23	66.42	21.67	42.48	61.33	49.67	48.31
	MemoryBank	AAAI'24	66.93	23.33	46.76	66.67	55.33	51.80
	Generative	UIST'23	75.00	26.67	51.83	68.33	51.67	54.70
	MetaGPT	ICLR'24	71.04	26.67	50.40	64.67	53.33	53.22
	ChatDev	ACL'24	67.91	23.33	44.81	61.33	55.00	50.48
	G-Memory	NeurIPS'25	77.61	26.67	48.73	69.72	50.67	54.68
	M-Memory	proposed	**84.33**	**35.00**	**55.43**	**73.67**	**63.00**	**62.29**

Example: Task You are in the middle of a room. Looking quickly around you, you see a bed 1, a desk 2, a desk 1, a drawer 6, a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a garbagecan 1, a laundryhamper 1, a safe 1, a shelf 6, a shelf 5, a shelf 4, a shelf 3, a shelf 2, and a shelf 1. Your task is to: look at bowl under the desklamp.
Trajectory:The agent manages to find a bowl 1 on desk 2 Act 14: think: Now I find a bowl (1) on desk 2. Next, I need to take it. Obs 14: OK. Act 15: take bowl 1 from desk 2 Obs 15: You pick up the bowl 1 from the desk 2. Act 16: think: Now I take a bowl (1). Next, I need to find a desklamp. A desklamp is more likely to appear in desk (1-2), shelf (1-6), or drawer (1-6). I can check one by one, starting with desk 1 since I already found a desklamp there. Obs 16: OK. Act 17: go to desk 1 Obs 17: On the desk 1, you see a desklamp 1, a keychain 3, a mug 2, a pen 2, a pen 1, and a pencil 2. Act 18: think: Now I find a desklamp (1) on desk 1. Next, I need to use it. Obs 18: OK. Act 19: use desklamp 1 Obs 19: You turn on the desklamp 1. reward: True
Output: Construct the overall approach: Find and take a bowl, then find and use a desklamp. Search for the Bowl: Check all locations systematically in order until the bowl is found (e.g., drawers, desks, shelves, garbage can, laundry hamper, and any other possible places). Find the Bowl: Locate the bowl on desk 2. Take the Bowl: Pick up the bowl from desk 2. Search for the Desklamp: Recall that a desklamp was found earlier on desk 1. Go to Desk 1: Move to desk 1 where the desklamp is located. Use the Desklamp: Turn on the desklamp.
Now it's your turn! Here is the task:
{task}
Successful Trajectory of Similar Task:
{trajectory}
Unsuccessful Trajectory of Similar Task:
{false_traj}
Successful Trajectory of Diverse Task:
{diverse_traj}
Output:

C.2 Task Relevance Evaluation

Task Relevance Evaluation

You are an agent designed to score the relevance between two pieces of text. You will be given a successful/unsuccessful case where you successfully/unsuccessfully complete the task. Then you will be given an ongoing task. Do not summarize these two cases, but rather evaluate how relevant and helpful the case is for the ongoing task, on a scale of 1-10.
Success/Failure Case:
{case}
Ongoing task:
{task}
Score:

C.3 Finetune Insights

Finetune Insights

You are an advanced reasoning agent that can add, edit or remove rules from your existing rule set, based on forming new critiques of past task trajectories. You will be given successful tasks trials/ fail tasks trials/ diverse successful tasks trials in which you were placed in a household environment and tasks to complete.
Requirements:
- Avoid vague statements; ensure each insight has a clear causal relationship.
- Focus only on strategies that apply to a broad range of scenarios rather than case-specific advice.
- Keep the language concise and to the point, ensuring clarity and practical value.
- The insights you summarize must follow the "XXX, because XXX" format. They should not mention specific items but should instead extract general success principles applicable to similar tasks. These insights must be enlightening and provide guidance for future problems.
Here are the trials: {history}
Here are the EXISTING RULES: {existing_rules}
The available operations are: **AGREE (if the existing rule is strongly relevant for the task), REMOVE (if one existing rule is contradictory or similar/duplicated to other existing rules), EDIT (if any existing rule is not general enough or can be enhanced, rewrite and improve it), ADD (add new rules that are very different from existing rules and relevant for other tasks). Each needs to closely follow their corresponding formatting below (any existing rule not edited, not agreed, nor removed is considered copied)**:
AGREE [EXISTING RULE NUMBER]: [EXISTING RULE]
REMOVE [EXISTING RULE NUMBER]: [EXISTING RULE]
EDIT [EXISTING RULE NUMBER]: [NEW MODIFIED RULE]
ADD: [NEW RULE]
Do not mention the trials in the rules because all the rules should be GENERALLY APPLICABLE. Each rule should be concise and easy to follow. Any operation can be used MULTIPLE times. Do at most 4 operations and each existing rule can only get a maximum of 1 operation.
By examining the successful trials/ the fail trials/ the diverse successful trials, and the list of existing rules, you can perform the following operations: add, edit, remove, or agree so that the new list of rules are general and high level insights of the successful trials/ the fail trials/ the diverse successful trials or proposed way of Thought so they can be used as helpful tips to different tasks in the future. Have an emphasis on tips that help the agent perform better Thought and Action.

C.4 Initialization of Agent Memory

Initialization of Agent Memory

You are a smart agent designed to solve problems. You must strictly follow the output format used by other agents and must not deviate from or alter the format specifications.
{agent_memory}

Below are some examples of similar tasks that were successfully completed. Please use these as references to guide your thinking and approach to the current task:
{few_shots}
Your Own Past Successes (Execution Patterns) Here are examples of successful execution processes you've previously used on similar tasks. Pay special attention to the step-by-step procedures and strategies, especially when encountering obstacles:
{memory_few_shots}
Key Insights from Related Successful Tasks The following are insights gathered during the execution of similar successful tasks. You may consider whether you can directly apply them during your task execution to improve problem-solving accuracy.
{Shortcuts}
Key Insights from Related Failed Tasks The following are the error-prone points gathered during the execution of similar failed tasks. You should be vigilant not to repeat them during your task execution to improve problem-solving accuracy.
{Pitfalls}
Key Insights from Diverse Successful Tasks The following are the generalized knowledge gathered during the execution of unsimilar successful tasks. You should be vigilant not to repeat them during your task execution to improve problem-solving accuracy.
{Cross Domain Consensus}
Your Turn: Take Action! Use the above examples and insights as a foundation, and now work on the following task:
Task description: {task_description}
Key steps: {key_steps}
Detailed trajectory: {trajectory}

C.5 Insight Summarization

Insight Summarization

You are an expert agent skilled at distilling generalizable knowledge from specific task executions. You are provided with the details of a newly completed task (including the query, execution status, and interaction trajectory) alongside a list of currently stored insights.
Your goal is to analyze this specific new execution and extract a new, high-level insight that captures the core reason for success or failure.
Guidelines: - Analyze the interaction trajectory to identify the key factors that led to the execution status.
- Generalize the specific experience from trace into a reusable rule or principle applicable to new tasks.
- Cross-reference the "currently stored insights" to ensure the new insight adds novelty and is not a duplicate of existing knowledge.
- Strictly base your summary on the provided execution trace; do not invent information.
Output Format: - Start your response directly with the numbered list, no preamble or explanations.
- If no significant new insight is found that isn't already covered, output nothing.
- Each insight should be a short, actionable sentence.
- Use the following format exactly:
1. Insight derived from the new execution
Inputs:
New Task Query:
{New_Task_Query}
Execution Status & Interaction Trajectory:
{Execution_Status_and_Trajectory}
Output:

D Related Works

Single-Agent Memory. Memory is a primary driver for agents to accumulate experience and explore their environment through interaction (Wang et al., 2024b; Xi et al., 2025; Gao et al., 2024; Li et al., 2024). It plays a pivotal role in LLM agents for both task solving and social simulation; this research focuses on the former. Early research on agent memory was confined to simple intra-trial memory, primarily addressing the context window limitations of LLMs in chatbot applications. Models such as MemoryBank (Zhong et al., 2024), ChatDB (Hu et al., 2023), MemoChat (Lu et al.,

2023), and MemGPT (Packer et al., 2023) typically employ a Retrieval-Augmented Generation (RAG) approach with similarity-based chunk retrieval. Subsequent developments have progressed towards more cognitively-inspired memory architectures, entailing: (1) an expansion of memory scope to cross-trial memory (e.g., ExpeL (Zhao et al., 2024), Synapse (Zheng et al., 2024)); (2) an extension of application domains to computer control (Zheng et al., 2024), embodied action (Zhu et al., 2023), scientific discovery (Tang et al., 2025), and programming reasoning (Shinn et al., 2023b); and (3) an evolution of management techniques from coarse-grained text similarity to more sophisticated abstraction and summarization of acquired knowledge, as exemplified by models like A-Mem (Xu et al., 2025), Mem0 (Chhikara et al., 2025), and MemInsight (Salama et al., 2025).

Multi-Agent System Memory. However, a significant research gap remains in memory mechanisms designed specifically for multi-agent systems (MAS). Some representative frameworks, such as LLM-Debate (Wang et al., 2024d) and hybrid agents (Wang et al., 2024a), omit a memory component entirely, while others employ only rudimentary intra-trial memory schemes (Qian et al., 2024b; Yin et al., 2023). Even frameworks that attempt cross-trial memory (Qian et al., 2024a) often reduce memory to an artifact of the final outcome, overlooking the nuanced interactions between agents. G-Memory (Zhang et al., 2025a) was introduced to address this gap, providing a hierarchical agent memory system for MAS that manages complex interactions via a three-layer graph structure and enables self-evolution by integrating new task experiences. However, G-Memory over-rely on similar successful tasks, neglecting the synergistic effects of diverse cognitive perspectives on execution and thus limiting the performance ceiling of the MAS.

LLM-based Multi-Agent Systems. Existing multi-agent systems (MAS) typically lack the capability for continuous evolution through interaction with their environment (Ou et al., 2025; Liang et al., 2024). Early frameworks such as AutoGen (Wu et al., 2024), Camel (Li et al., 2023), and AgentVerse (Chen et al., 2023) rely exclusively on predefined workflows. Recent work (Hu et al., 2024b; Zhang et al., 2024a; Yuan et al., 2025; Hu et al., 2024a; Zhang et al., 2024b) has introduced a degree of adaptivity by generating dynamic MAS that respond to environmental feedback. However, such evolution is often a one-shot process. For instance, AFlow (Zhang et al., 2024b) employs Monte Carlo Tree Search to construct a task-specific MAS, yet it lacks the capacity to evolve with increased task exposure or transfer knowledge across domains (Zhang et al., 2025b).