BEYOND POSITIVE REINFORCEMENT: A HIERARCHICAL MULTI-PERSPECTIVE MEMORY FRAMEWORK FOR MULTI-AGENT SYSTEMS

Anonymous authors
Paper under double-blind review

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 design consideration remains insufficiently addressed in existing approaches: they predominantly rely on positive reinforcement, with insufficient utilization of the pedagogical value provided by failure experiences. 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 draws inspiration from three empirically-grounded memory organization strategies. Concretely, M-Memory integrates both successful and failed memory records and structures them at two levels. At the high level, the framework 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).

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, predefined 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. As shown in Fig 1, 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.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To address the limitations, we propose the Multi-Perspective Memory (M-Memory), a comprehensive memory framework for MAS that integrates both successful experiences and lessons from failure. This framework organizes system memory into two levels: a high-level insight memory and a low-level interaction memory, each structured around three cognitive perspectives. At the high level, M-Memory extracts abstract knowledge through three complementary perspectives, each of which is derived from corresponding neural research: **Shortcuts** extracts reusable procedural templates from similar successful tasks, drawing motivation from research on *hippocampal episodic retrieval* Tulving et al. (1972), which demonstrates how specific past episodes can be recombined to guide future behavior, thereby enhancing execution efficiency for specific task archetypes; **Pitfalls** identifies critical failure modes from similar failed tasks, reflecting *amygdala-driven avoidance learning* for evading known failure modes, to bolster system robustness through proactive risk mitigation Mowrer (1947); Holroyd & Coles (2002); Baumeister et al. (2001); **Cross-domain Consensus** abstracts generalizable principles from successful tasks across disparate domains, drawing motivation from research on *analogical transfer* Gentner (1983); Badre & Nee (2018), which shows how structural similarities across domains can support the extraction of transferable knowledge, thereby improving generalization and fostering innovation. At the low level, the interaction memory enhances trajectory summarization for critical steps. Rather than drawing solely from successful trajectories, the summarization process also incorporates interaction traces from similar failed tasks as counterexamples and from disparate successful tasks as sources for analogical transfer. 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 four 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 four 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 A.7.

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.

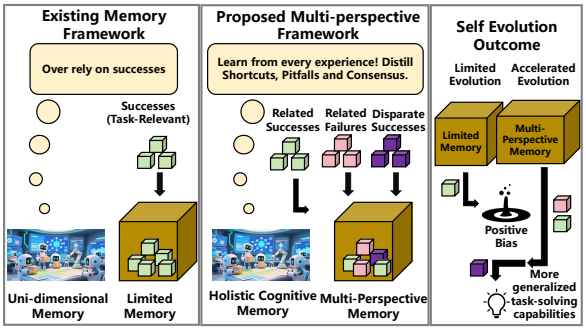

Figure 1: This figure contrasts two memory frameworks by their evolutionary outcomes: (i) conventional uni-dimensional frameworks that over-rely on analogous successes, proving susceptible to positive bias and iterative failures; and (ii) our proposed multi-perspective framework, which holistically processes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

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 $\Psi_i \in \{\text{Solved, Failed}\}$, 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 support set $\Omega_i \subseteq \mathcal{Q}$ whose evidence type is perspective-specific: for $\iota_i \in \mathcal{I}_S \cup \mathcal{I}_C$, Ω_i collects the queries successfully executed under κ_i, whereas for $\iota_i \in \mathcal{I}_P$, Ω_i is initialized by Eq.13 from the failed query that distilled κ_i and subsequently collects the queries that encountered or avoided the recorded failure mode. 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:

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

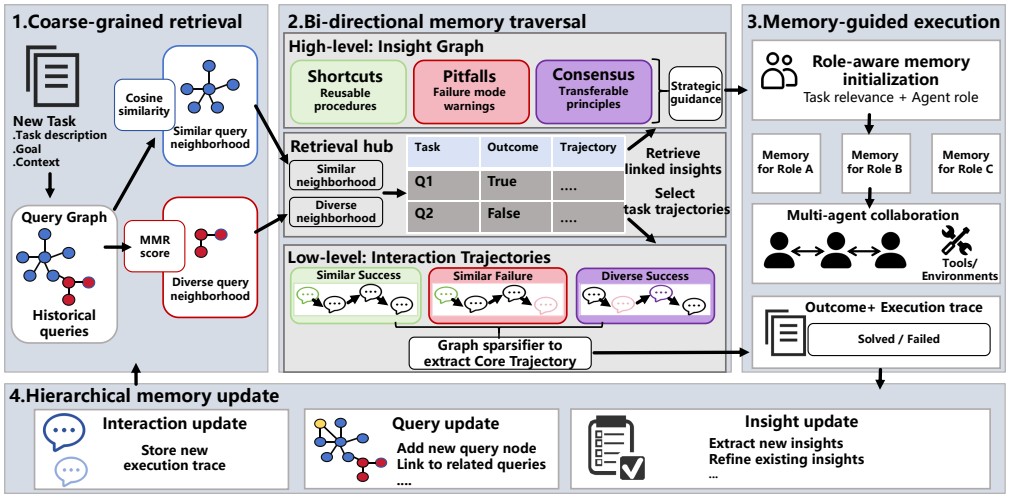

Figure 2: Overview of M-Memory. For a new task, cosine similarity and Maximal Marginal Relevance (MMR) respectively retrieve similar and diversity-oriented query neighborhoods, each expanded over two hops. Anchored on these queries, M-Memory traverses the Query Graph bidirectionally: upward traversal collects high-level insights (Shortcuts, Pitfalls, and Cross-domain Consensus), while downward traversal selects low-level interaction traces from similar successes, similar failures, and diverse successes. An LLM-based graph sparsifier distills a compact core trajectory, using failure and diverse-success traces as auxiliary references. These two levels jointly initialize role-aware agent memories for collaborative execution. Afterwards, both successful and failed experiences are stored, and outcome-conditioned updates refine the query and insight structures, closing a self-evolving loop.

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 & 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:

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

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:

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

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 cooper-

ation (detailed prompts are in Appendix A.5.1):

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

where $\text{Rel}(q, q_i, x)$ (Appendix A.5.2) 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.

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 A.6.1):

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

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.

Notably, when certain memory perspectives yield empty results, M-Memory employs a perspective degradation mechanism, proceeding with only the remaining available perspectives. For instance, during the cold-start phase when all insight sets are empty, the system executes the task directly.

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:

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

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:

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

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

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 $\mathcal{Q}_M \subseteq \tilde{\mathcal{Q}}^S$ 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 $\mathcal{I}$ (equation 2) with the new query node Q_{new}:

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

Otherwise, no update is applied to any support set: each Ω_i remains a single-type, attributable-evidence statistic (success validation for $\iota_i \in \mathcal{I}_S \cup \mathcal{I}_C$, and failure-mode engagement with a resolved outcome for $\iota_i \in \mathcal{I}_P$). Decrementing Ω_i on a reuse failure would conflate falsification with absent validation, because such a failure is not attributable to κ_i: it may stem from execution-level

deviations, environmental stochasticity, or incidental inter-agent errors rather than from the guiding insight itself (Details can see Appendix A.3.1).

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 A.6.2) 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 $\mathcal{I}$:

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

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

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

Subsequently, the corresponding insights in $\mathcal{I}$ (Eq. 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 A.6):

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

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 HotpotQAYang 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.1.1. **Baselines.** For comparative analysis, we select three 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); three extensively used multi-agent memory architectures: MetaGPT Hong et al. (2024), ChatDev Qian et al. (2024a), and G-Memory Zhang et al. (2025a); and two memory architectures that explicitly exploit failure experiences: Reflexion Shinn et al. (2023a) and RLEM Zhang et al. (2023) (details are provided in Appendix A.1.2). **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.1.3). 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 4. The number of hop expansion in Eq. 4 is selected as 2. To rigorously assess long-term self-evolution, inspired by Reflexion Shinn et al. (2023a), we evaluate all methods equipped with a memory module over three consecutive rounds on the identical task set, presented in a fixed task order. Within each round, the memory is updated online upon the completion of every task, and the accumulated memory is carried over unchanged to the next round. Round 1 therefore isolates online memory construction from scratch, whereas the subsequent rounds measure the gains attributable to cross-round retention. We report the results at the two endpoints separately: Tables 1 and 4 present the Round-1 (cold-start) results, and Tables 2 and 5 present the post-evolution results, defined as the Round-3 score, at which point the memory has accumulated a full two rounds of prior experience. Round-to-round variability is instead controlled by repeating the entire three-round protocol three times, and we report round-level standard deviations together with significance tests (Appendix A.2.2).

Table 1: Round-1 (cold-start) performance with the Qwen-2.5-14b backbone. Compared to the best-performing baseline of the same round, M-Memory achieves an average improvement of 3.70% with AutoGen, 3.80% with DyLAN, and 5.05% with MacNet.

MAS	Method	Venue	Alfworld	PDDL	PDDL(RE)	FEVER	HotpotQA	Avg.
AutoGen	No-memory	–	79.10±2.24	25.00±4.41	45.17±3.68	62.33±3.21	51.33±2.52	52.59
	Voyager	NeurIPS'23	80.85±2.15	26.67±3.33	52.25±3.31	63.33±2.89	49.67±3.79	54.55
	MemoryBank	AAAI'24	76.87±1.49	18.33±3.33	43.17±2.85	64.67±3.51	51.33±2.31	50.87
	Generative	UIST'23	82.09±1.49	30.56±4.19	48.62±3.53	62.33±3.79	47.67±3.51	54.25
	MetaGPT	ICLR'24	80.10±1.55	28.33±3.33	49.94±2.65	63.33±3.51	54.67±2.52	55.27
	ChatDev	ACL'24	74.63±2.99	18.33±2.89	39.04±3.54	64.33±2.31	53.33±4.16	49.93
	Reflexion	NeurIPS'23	78.36±1.88	23.33±4.41	43.28±3.34	63.00±1.73	51.00±2.33	51.79
	RLEM	NeurIPS'23	79.10±2.59	20.00±3.33	44.57±3.53	62.67±2.89	54.00±3.00	52.07
	G-Memory	NeurIPS'25	81.34±1.49	31.67±3.33	53.78±3.70	67.33±2.08	60.33±3.51	58.89
	M-Memory	proposed	**83.58**±1.97	**36.67**±1.67	**58.38**±2.77	**69.67**±1.15	**64.67**±1.53	**62.59**
DyLAN	No-memory	–	76.87±1.29	23.33±2.89	48.61±2.70	58.67±3.21	49.33±3.79	51.36
	Voyager	NeurIPS'23	80.60±1.97	25.00±1.67	49.38±2.79	52.33±2.08	50.67±3.06	51.60
	MemoryBank	AAAI'24	77.61±2.24	18.33±6.01	44.81±3.34	59.00±3.00	45.67±2.08	49.08
	Generative	UIST'23	77.86±2.62	23.33±3.33	47.65±2.93	50.33±2.08	51.00±3.61	50.03
	MetaGPT	ICLR'24	79.35±2.40	28.33±1.67	54.56±3.50	57.67±3.51	52.67±2.52	54.52
	ChatDev	ACL'24	75.37±2.99	25.00±2.89	47.66±3.02	55.33±1.15	53.67±4.51	51.41
	Reflexion	NeurIPS'23	75.37±1.29	26.67±2.89	45.54±2.72	62.33±2.52	45.00±2.33	50.98
	RLEM	NeurIPS'23	77.61±2.59	23.33±5.00	46.76±3.69	65.33±2.89	53.33±2.52	53.27
	G-Memory	NeurIPS'25	80.10±2.28	30.00±2.89	49.99±2.62	65.33±3.06	42.67±3.06	53.62
	M-Memory	proposed	**82.09**±1.29	**31.67**±4.41	**56.19**±3.18	**72.33**±3.06	**49.33**±2.89	**58.32**
MacNet	No-memory	–	61.44±2.40	13.33±3.33	22.95±3.13	62.33±3.06	49.33±2.31	41.88
	Voyager	NeurIPS'23	61.94±3.42	21.67±1.67	42.48±3.36	61.33±3.79	49.67±2.52	47.42
	MemoryBank	AAAI'24	66.92±3.53	20.56±2.55	44.76±3.75	63.67±2.08	53.33±2.31	49.85
	Generative	UIST'23	67.66±1.72	16.67±1.67	31.83±3.25	64.33±3.06	51.67±2.89	46.43
	MetaGPT	ICLR'24	65.92±3.37	26.67±1.67	50.40±2.73	64.67±6.03	53.33±3.06	52.20
	ChatDev	ACL'24	67.91±1.29	18.33±1.67	34.81±2.83	63.33±3.06	53.33±2.08	47.54
	Reflexion	NeurIPS'23	65.67±2.99	21.67±2.89	45.00±3.06	58.33±2.89	52.67±4.04	48.67
	RLEM	NeurIPS'23	61.94±2.69	20.00±2.89	41.28±4.03	66.00±2.65	44.33±2.89	46.71
	G-Memory	NeurIPS'25	71.14±2.83	26.67±5.00	47.34±3.42	68.33±2.52	47.67±2.08	52.23
	M-Memory	proposed	**74.63**±2.24	**31.67**±5.00	**53.45**±3.77	**67.67**±4.16	**59.00**±3.46	**57.28**

Table 2: Post-evolution performance with the Qwen-2.5-14b backbone. Compared to the best-performing baseline under the same protocol, M-Memory achieves an average improvement of 7.89% with AutoGen, 8.97% with DyLAN, and 7.57% with MacNet.

MAS	Method	Venue	Alfworld	PDDL	PDDL(RE)	FEVER	HotpotQA	Avg.
AutoGen	No-memory	–	80.10±2.40	27.78±2.55	45.81±2.79	64.33±2.52	51.33±3.06	53.87
	Voyager	NeurIPS'23	80.85±0.86	31.67±1.67	55.55±1.84	64.33±2.08	49.67±2.08	56.41
	MemoryBank	AAAI'24	77.61±1.29	20.56±2.55	44.75±2.07	65.67±2.31	51.33±1.15	51.98
	Generative	UIST'23	83.58±1.29	31.67±3.33	50.81±1.91	63.33±2.52	49.00±1.73	55.68
	MetaGPT	ICLR'24	80.10±1.14	27.78±1.92	52.49±2.34	63.33±1.15	54.67±1.15	55.67
	ChatDev	ACL'24	76.87±1.97	20.56±1.92	40.47±1.72	64.67±1.15	57.33±2.31	51.98
	Reflexion	NeurIPS'23	85.82±1.29	33.33±2.89	49.14±2.47	65.33±1.00	56.00±1.73	57.92
	RLEM	NeurIPS'23	80.60±1.29	21.67±2.89	41.71±2.54	63.33±2.08	58.67±1.53	53.20
	G-Memory	NeurIPS'25	85.07±1.29	33.33±1.67	55.43±2.63	70.33±1.15	62.33±3.21	61.30
	M-Memory	proposed	**91.04**±0.75	**43.33**±3.33	**64.22**±2.67	**78.67**±0.58	**68.67**±3.06	**69.19**
DyLAN	No-memory	–	76.87±1.49	24.44±3.47	47.92±3.64	59.33±2.08	53.33±2.31	52.38
	Voyager	NeurIPS'23	82.59±0.86	26.11±3.47	49.83±2.60	65.33±1.53	53.67±1.53	55.51
	MemoryBank	AAAI'24	78.36±1.49	18.33±1.67	44.03±2.16	58.67±2.31	55.33±1.53	50.94
	Generative	UIST'23	80.10±0.86	25.56±0.96	50.23±2.50	60.00±1.00	49.33±2.08	53.04
	MetaGPT	ICLR'24	80.85±1.14	27.78±0.96	53.13±2.46	61.67±1.53	53.33±2.08	55.35
	ChatDev	ACL'24	75.12±1.55	25.00±1.67	51.23±2.14	60.33±2.89	53.67±2.08	53.07
	Reflexion	NeurIPS'23	85.07±1.49	31.67±2.89	49.41±2.48	63.33±1.53	49.33±1.53	55.76
	RLEM	NeurIPS'23	79.10±1.49	26.67±4.41	48.44±2.51	67.67±1.15	56.33±2.08	55.64
	G-Memory	NeurIPS'25	83.58±0.75	30.00±1.67	51.23±1.97	66.33±2.08	55.33±2.08	57.29
	M-Memory	proposed	**87.31**±1.29	**41.67**±2.89	**66.96**±1.81	**75.67**±1.53	**59.67**±1.53	**66.26**
MacNet	No-memory	–	61.19±2.24	13.33±2.89	22.95±3.27	65.33±2.52	53.33±3.06	43.23
	Voyager	NeurIPS'23	66.42±1.97	21.67±1.67	42.48±2.34	61.33±1.15	49.67±2.52	48.31
	MemoryBank	AAAI'24	66.92±0.86	23.33±3.33	46.76±2.02	66.67±1.53	55.33±1.15	51.80
	Generative	UIST'23	75.12±0.86	26.67±1.67	51.83±2.51	68.33±1.15	51.67±1.53	54.72
	MetaGPT	ICLR'24	71.14±1.88	26.67±2.89	50.40±1.89	64.67±2.08	53.33±1.53	53.24
	ChatDev	ACL'24	67.91±1.49	23.33±1.67	44.81±1.87	61.33±2.08	55.00±1.00	50.48
	Reflexion	NeurIPS'23	67.91±1.29	25.00±2.89	49.19±2.57	67.33±1.15	54.00±1.00	52.69
	RLEM	NeurIPS'23	67.16±1.29	30.00±2.89	50.37±1.77	68.67±1.53	55.33±2.31	54.31
	G-Memory	NeurIPS'25	77.61±1.49	26.67±3.33	48.73±2.64	69.67±2.52	50.67±1.53	54.67
	M-Memory	proposed	**84.33**±1.29	**35.00**±3.33	**55.43**±2.08	**73.67**±2.08	**63.00**±2.00	**62.29**

4.2 Main Results

Tables 1 and 4 report the performance of all methods when the memory store is initialized as empty. Even without any prior insight, M-Memory surpasses the strongest baseline by an average of 4.69%, 4.52% and 1.24% on AutoGen, DyLAN and MacNet under gpt-4o-mini, and by 3.70%, 3.80% and 5.05% under Qwen-2.5-14b. Since all methods observe exactly the same task stream within a round, this gap can be attributed solely to the hierarchical organization and the multi-perspective retrieval that M-Memory employs while constructing memory across rounds. Tables 2 and 5 further report the Round-3 scores, that is, the performance after memory has accumulated over two full rounds. Under gpt-4o-mini the margins over the strongest baseline are 3.18%, 7.61% and 6.95%, and under Qwen-2.5-14b they are 7.89%, 8.97% and 7.57%. The margin grows in five of the six settings, indicating that M-Memory converts accumulated experience into further gains more effectively than the baselines and therefore exhibits a stronger self-evolution capability. Moreover, among the baselines

Table 3: 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
	Shortcuts Only	83.96	33.33	52.93	65.33	61.33	59.38
	Shortcuts + Pitfalls	85.82	39.17	59.93	74.67	62.67	64.45
	Complete M-Memory	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
	Shortcuts Only	80.97	30.00	54.42	66.67	55.33	57.48
	Shortcuts + Pitfalls	84.33	36.67	60.08	69.33	55.67	61.22
	Complete M-Memory	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
	Shortcuts Only	76.37	28.33	51.10	64.67	51.33	54.36
	Shortcuts + Pitfalls	82.84	31.67	54.01	72.33	61.33	60.44
	Complete M-Memory	84.33	35.00	55.43	73.67	63.00	62.29

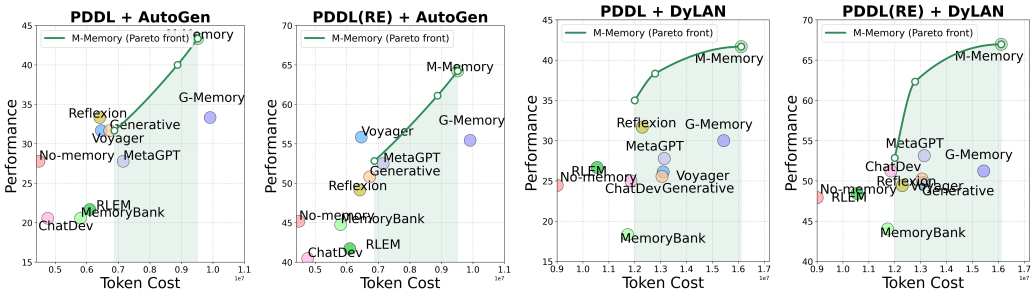

Figure 3: The relationship between performance and total system token cost across different memory architectures. M-Memory achieves higher-performance collective memory while maintaining a stable and controlled token cost. The LLM backbone is Qwen-2.5-14b.

that explicitly exploit failure experience, M-Memory achieves better performance under both the empty-memory and the accumulated-memory settings. We attribute this gap to how each method resolves the trade-off between memory granularity and transferability. Reflexion distills fine-grained causal hypotheses from failures on a single task, yet these hypotheses remain tightly coupled to the concrete execution trajectory and therefore transfer poorly to similar tasks, yielding only marginal gains. RLEM does transfer knowledge across tasks, but its numeric encoding of task steps is meaningful only among homogeneous tasks and loses its semantics once the task domain changes, so the transfer breaks down. M-Memory instead decouples fine-grained trajectory memory from distilled insight memory through bidirectional memory traversal, which allows abstract insights to be reused across tasks while the underlying details are preserved intact, thereby improving transferability without sacrificing fidelity. In addition, its multi-perspective memory mechanism retrieves and injects knowledge originating from other domains, going beyond the homogeneous-task regime and raising the upper bound of attainable performance. Notably, we further apply Welch's t-test to examine whether the Round-3 performance difference between M-Memory and G-Memory is statistically significant, and the results confirm that M-Memory outperforms or matches the G-Memory baseline in all tested conditions (Appendix A.2.2).

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 under the Round-3 post-evolution protocol (Table 3). 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 9.81%, 8.78%, and 7.93%, respectively, relative to the full three-perspective configuration. Even when both successful and failed insights are utilized, performance remains suboptimal, exhibiting average drops of 4.74%, 5.04%, 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. To provide a more concrete illustration of the performance gains contributed by each memory perspective, we present several case studies in the Appendix A.4, demonstrating how the Pitfall and Consensus perspectives, respectively, shape task execution within the MAS framework.

Cost Analysis. To evaluate the token efficiency of M-Memory, Fig. 3 characterizes the trade-off between task performance and token cost across different memory architectures. As shown in Fig. 3, M-Memory attains the largest performance gains under both the PDDL+AutoGen and PDDL(RE)+AutoGen configurations while incurring an additional token cost of only 5.0×10^6. GMemory, in contrast, yields markedly smaller improvements at a higher additional cost of 7.1×10^6. To further probe the performance boundary of M-Memory under varying token budgets, we trace its Pareto-optimal frontier by reducing the number of memory items retrieved during task execution and by lowering the frequency of insight refinement. Taken together, these results demonstrate that M-Memory establishes higher-performing collective memory while keeping the token cost stable and controllable, thereby achieving superior token efficiency.

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 4. 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 $k = 4$ throughout our experiments.

4.4 Case Study

In this section, we will discuss how M-Memory synthesizes multi-dimensional memory cues to guide complex task executions. Taking the PDDL environment as an example, upon receiving a task, M-Memory retrieves relevant knowledge and decomposes it into three complementary perspectives to guide decision-making: First, the **Cross-domain Consensus** provides a high-level assembly strategy, explicitly advising the agent to prioritize 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** 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** 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 integrates the multi-perspective insights with core execution trajectories distilled from past tasks to facilitate task execution, thereby significantly improving the execution efficiency of the current task. Other case studies can see Figure 5.

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.

6 AI USE STATEMENT

The authors declare that Generative AI tools were used during the preparation of this work for language editing and proofreading purposes. The use of AI was strictly limited to improving the clarity, grammar, and readability of the text. No AI tools were used for generating core concepts, experimental design, data analysis, or result interpretation. The authors take full responsibility for the content and accuracy of the final manuscript.

REFERENCES

D. Badre and D. E. Nee. Frontal cortex and the hierarchical control of behavior. *Trends in cognitive sciences*, 22(2):170–188, 2018.

R. F Baumeister, E. Bratslavsky, C. Finkenauer, and K. D Vohs. Bad is stronger than good. *Review of general psychology*, 5(4):323–370, 2001.

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

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

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

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

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

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

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

D. Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2): 155–170, 1983.

C. B Holroyd and M. G. Coles. The neural basis of human error processing: Reinforcement learning, dopamine, and the error-related negativity. *Psychological review*, 109(4):679, 2002.

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

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

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

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

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

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

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

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

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

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

O Mowrer. On the dual nature of learning—a re-interpretation of" conditioning" and" problem-solving.". *Harvard educational review*, 1947.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

E. Tulving et al. Episodic and semantic memory. *Organization of memory*, 1(381-403):1, 1972.

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

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

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

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

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

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

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

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

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

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

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

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

D. Zhang, L. Chen, S. Zhang, H. Xu, Z. Zhao, and K. Yu. Large language models are semi-parametric reinforcement learning agents. *Advances in Neural Information Processing Systems*, 36:78227–78239, 2023.

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

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

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

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

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

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

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

X. Zhu, Y. Chen, H. Tian, C. Tao, W. Su, C. Yang, G. Huang, B. Li, L. Lu, X. Wang, et al. 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*, 2023.

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

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

A Appendix

A.1 Experimental Details

A.1.1 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.1.2 Baselines

For comparative analysis, we select a rigorously curated set of baseline methods as evaluation benchmarks, which represent milestone achievements in the literature of this field. 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 constructs a cognitively grounded long-term memory module featuring dynamic summarization, reflection, and temporal retrieval; and *Generative Agents* Park et al. (2023), which employs a continuous memory stream integrated with observation–reflection–planning loops to drive autonomous behavior. Furthermore, we incorporate three extensively validated multi-agent architectures: *MetaGPT* Hong et al. (2024), which orchestrates inter-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 adopts structured memory representations to facilitate efficient cross-agent knowledge sharing and state-aware reasoning. In addition, we select two memory architectures that explicitly exploit failure experiences: *Reflexion* Shinn et al. (2023a), which translates task failure experiences into verbalized improvement strategies stored in memory, subsequently retrieving and injecting them into prompts for similar tasks, thereby enabling policy iterative refinement through linguistic feedback rather than parameter updates; and *RLEM* Zhang et al. (2023), which enforces an explicit division of labor between parametric knowledge and failure memory: the LLM acts solely as the policy that proposes actions in a given state, while all learning is delegated to a tabular Q-function, so that no gradient updates to the model parameters are required. Each entry of the Q-table is indexed by a (task goal, observation) pair and stores, for every action attempted in that state, a running value estimate together with its visit count. Once an interaction step yields a successor state, the corresponding entry is updated with an n-step bootstrapped target and its visit count is incremented accordingly. Notably, the verbal reflections produced by Reflexion are tightly coupled to the specifics of the failure task that generated them, and are therefore transferable only when the same task is revisited. Under our evaluation protocol, in which agents execute a sequence of distinct tasks round by round, the reflections accumulated in earlier rounds no longer apply to later ones and can even be actively misleading; as a result, Reflexion performs worse in subsequent rounds than in the initial one. To let it realize its full potential, we accordingly adapt the execution protocol for Reflexion: instead of rotating across tasks, the same task is executed three times consecutively with memory retained, so that every reflection remains applicable to the task from which it was derived. To verify the reproducibility of GMemory, we conducted multiple replication experiments using the open-source code released by the authors. Under the original configuration with gpt-4o-mini as the backbone model, the method fails to reproduce the performance reported in the original paper on certain datasets, yielding inconsistent replication results overall. Notably, when the backbone model is replaced with Qwen2.5-14B, GMemory not only reproduces the reported results reliably but even surpasses them on several datasets. We therefore conjecture that the replication discrepancy observed on gpt-4o-mini stems primarily from behavioral drift induced by updates to the backbone model's API version, rather than from any flaw in our replication procedure. We are also in ongoing discussions with the authors to identify the precise cause.

A.1.3 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.

Table 4: Round-1 (cold-start) performance with the gpt-4o-mini backbone. Every memory architecture starts from an empty memory and learns online within the round; no prior insights are available. Compared to the best-performing baseline of the same round, M-Memory achieves an average improvement of 4.69% with AutoGen, 4.52% with DyLAN, and 1.24% with MacNet.

MAS	Method	Venue	Alfworld	PDDL	PDDL(RE)	FEVER	HotpotQA	Avg.
AutoGen	No-memory	–	70.65±2.15	4.44±0.96	19.68±2.25	58.67±2.31	48.33±4.04	40.35
	Voyager	NeurIPS'23	76.37±2.40	7.22±1.92	26.06±3.15	60.33±4.51	43.33±2.31	42.66
	MemoryBank	AAAI'24	70.65±3.11	2.22±0.96	11.04±2.49	61.33±2.89	46.33±3.79	38.31
	Generative	UIST'23	82.34±1.72	2.22±1.92	11.80±2.61	60.33±2.89	51.67±2.31	41.67
	MetaGPT	ICLR'24	81.84±2.83	7.78±1.92	20.88±2.73	61.33±3.06	54.33±3.06	45.23
	ChatDev	ACL'24	68.91±3.02	3.33±1.67	17.20±2.37	58.33±4.04	52.67±3.06	40.09
	Reflexion	NeurIPS'23	82.84±1.49	5.56±0.96	20.28±2.70	62.33±2.52	53.33±2.52	44.87
	RLEM	NeurIPS'23	79.10±2.40	3.89±3.47	14.50±2.50	59.00±2.65	52.33±2.52	41.76
	G-Memory	NeurIPS'25	80.85±2.62	4.44±1.92	22.29±2.97	63.33±3.06	40.33±2.52	42.25
	M-Memory	proposed	**83.58**±2.30	**11.67**±1.67	**29.71**±3.11	**66.33**±3.10	**58.33**±2.10	**49.92**
DyLAN	No-memory	–	70.40±2.62	1.11±0.96	6.58±1.98	58.67±2.52	48.33±3.51	37.02
	Voyager	NeurIPS'23	74.38±2.40	3.89±1.92	17.98±1.55	62.33±1.15	43.33±2.52	40.38
	MemoryBank	AAAI'24	60.20±1.88	3.89±2.55	18.38±2.24	59.67±3.06	46.67±2.08	37.76
	Generative	UIST'23	79.35±1.72	3.89±0.96	19.10±1.90	63.33±2.52	51.67±3.06	43.47
	MetaGPT	ICLR'24	80.35±2.40	7.78±0.96	22.75±2.47	60.33±3.79	54.33±2.89	45.11
	ChatDev	ACL'24	60.70±2.40	2.78±0.96	13.71±1.03	61.33±3.21	54.67±2.89	38.64
	Reflexion	NeurIPS'23	59.95±2.28	3.89±3.85	12.85±1.20	64.33±3.21	44.67±3.06	37.14
	RLEM	NeurIPS'23	65.67±2.59	4.44±3.47	13.14±1.27	67.33±2.89	57.33±2.52	41.58
	G-Memory	NeurIPS'25	76.37±2.15	2.78±1.92	12.57±1.13	65.33±3.21	46.33±3.21	40.68
	M-Memory	proposed	**79.10**±3.20	**10.56**±5.85	**28.84**±3.22	**69.33**±0.60	**60.33**±2.10	**49.63**
MacNet	No-memory	–	58.46±2.28	7.22±2.55	24.15±3.39	59.33±2.89	43.33±2.89	38.50
	Voyager	NeurIPS'23	68.91±3.11	10.56±2.55	19.62±2.66	60.67±3.21	45.67±4.73	41.09
	MemoryBank	AAAI'24	54.23±2.15	7.78±2.55	30.48±3.33	54.33±2.31	48.67±4.04	39.10
	Generative	UIST'23	67.66±2.15	5.56±1.92	18.40±2.46	52.33±2.89	47.33±3.21	38.26
	MetaGPT	ICLR'24	77.11±2.40	12.22±2.55	36.78±3.38	65.33±4.51	53.33±3.79	48.95
	ChatDev	ACL'24	59.45±2.62	2.78±2.55	18.80±2.48	61.33±1.53	51.67±2.89	38.81
	Reflexion	NeurIPS'23	58.96±2.59	7.78±3.47	27.31±2.89	51.33±3.06	49.33±2.89	38.94
	RLEM	NeurIPS'23	56.47±2.83	6.67±1.67	19.21±3.21	64.33±2.31	42.00±3.00	37.74
	G-Memory	NeurIPS'25	66.92±2.28	4.44±2.55	15.39±2.41	63.33±2.31	50.33±3.51	40.08
	M-Memory	proposed	**76.12**±1.30	**15.00**±2.89	**37.18**±3.76	**66.33**±7.80	**56.33**±2.90	**50.19**

Table 5: Post-evolution performance with the gpt-4o-mini backbone; memory is persistent across rounds and initialized from the experience accumulated in Rounds 1–2. Compared to the best-performing baseline under the same protocol, M-Memory achieves an average improvement of 3.18% with AutoGen, 7.61% with DyLAN, and 6.95% with MacNet.

MAS	Method	Venue	Alfworld	PDDL	PDDL(RE)	FEVER	HotpotQA	Avg.
AutoGen	No-memory	–	70.65±2.15	7.78±1.92	22.40±2.93	61.67±3.21	51.33±1.15	42.77
	Voyager	NeurIPS'23	77.36±1.55	7.22±0.96	27.08±1.84	61.33±1.53	44.67±2.31	43.53
	MemoryBank	AAAI'24	72.64±1.72	2.22±0.96	11.95±1.12	62.33±1.53	47.33±2.08	39.29
	Generative	UIST'23	84.83±1.55	7.22±1.92	25.59±1.87	63.33±2.08	54.33±2.08	47.06
	MetaGPT	ICLR'24	82.34±0.86	7.78±0.96	31.19±2.29	61.67±1.53	54.33±2.31	47.46
	ChatDev	ACL'24	70.40±1.55	4.44±0.96	18.63±1.45	59.67±1.15	53.67±2.52	41.36
	Reflexion	NeurIPS'23	83.33±0.86	10.00±3.33	23.99±1.93	62.67±2.08	54.33±1.53	46.86
	RLEM	NeurIPS'23	79.85±1.49	5.56±0.96	15.29±1.96	59.00±1.00	55.67±1.53	43.07
	G-Memory	NeurIPS'25	83.33±1.72	16.67±1.67	37.04±2.65	66.33±0.58	57.33±2.08	52.14
	M-Memory	proposed	**86.57**±1.49	**16.67**±2.89	**38.69**±2.22	**66.33**±2.31	**68.33**±1.53	**55.32**
DyLAN	No-memory	–	72.89±2.62	3.33±1.67	9.30±2.15	61.33±4.16	51.67±2.89	39.70
	Voyager	NeurIPS'23	75.12±0.86	1.11±0.96	9.00±1.24	63.33±2.52	44.33±1.53	38.58
	MemoryBank	AAAI'24	60.95±1.88	2.22±1.92	19.29±2.12	60.67±0.58	47.67±2.08	38.16
	Generative	UIST'23	81.84±1.14	3.89±0.96	11.89±1.26	66.67±2.08	54.33±2.52	43.72
	MetaGPT	ICLR'24	80.85±1.88	7.78±2.55	31.06±2.34	60.33±2.52	54.67±1.53	46.94
	ChatDev	ACL'24	61.69±2.83	2.78±0.96	15.14±1.02	62.67±1.53	56.33±1.53	39.72
	Reflexion	NeurIPS'23	83.33±0.86	11.11±0.96	19.15±1.17	64.67±1.53	42.67±1.53	44.19
	RLEM	NeurIPS'23	74.13±1.55	6.11±0.96	19.07±1.42	70.67±1.53	60.00±2.00	46.00
	G-Memory	NeurIPS'25	78.86±2.28	10.56±0.96	25.03±2.31	65.67±1.15	56.33±1.15	47.29
	M-Memory	proposed	**84.58**±1.88	**11.67**±3.33	**39.58**±2.19	**74.67**±0.58	**64.00**±1.00	**54.90**
MacNet	No-memory	–	63.18±1.88	7.22±2.55	24.15±2.26	64.33±3.21	48.33±3.51	41.44
	Voyager	NeurIPS'23	71.14±1.72	12.78±1.92	31.57±2.52	62.67±1.15	47.67±2.31	45.17
	MemoryBank	AAAI'24	55.72±2.28	9.44±0.96	32.22±2.21	55.67±2.52	50.33±1.53	40.68
	Generative	UIST'23	72.89±1.14	9.44±1.92	33.73±1.77	57.33±1.15	52.33±1.15	45.14
	MetaGPT	ICLR'24	77.36±1.88	12.22±0.96	36.78±2.09	65.67±1.53	53.67±1.53	49.14
	ChatDev	ACL'24	62.19±1.72	2.78±1.92	18.80±1.75	63.67±2.52	51.67±2.08	39.82
	Reflexion	NeurIPS'23	67.16±1.49	9.44±2.55	32.59±1.81	64.00±1.73	53.00±1.00	45.24
	RLEM	NeurIPS'23	65.17±1.72	16.67±4.41	32.97±1.59	65.00±1.00	56.00±1.73	47.16
	G-Memory	NeurIPS'25	72.14±1.88	4.44±1.92	20.56±1.86	63.33±1.53	58.33±1.15	43.76
	M-Memory	proposed	**84.33**±0.75	**20.00**±1.67	**38.81**±2.49	**72.00**±3.61	**65.33**±2.08	**56.09**

A.2 Experimental Result

A.2.1 Experimental Results with gpt-4o-mini

Table 4 reports the cold-start results and Table 5 reports the post-evolution results for different methods with the gpt-4o-mini backbone.

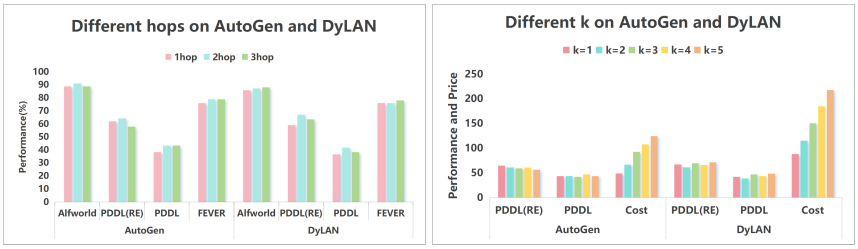

Figure 4: (a) Sensitivity analysis of the hop expansion; (b) Sensitivity analysis of the number of selected queries k.

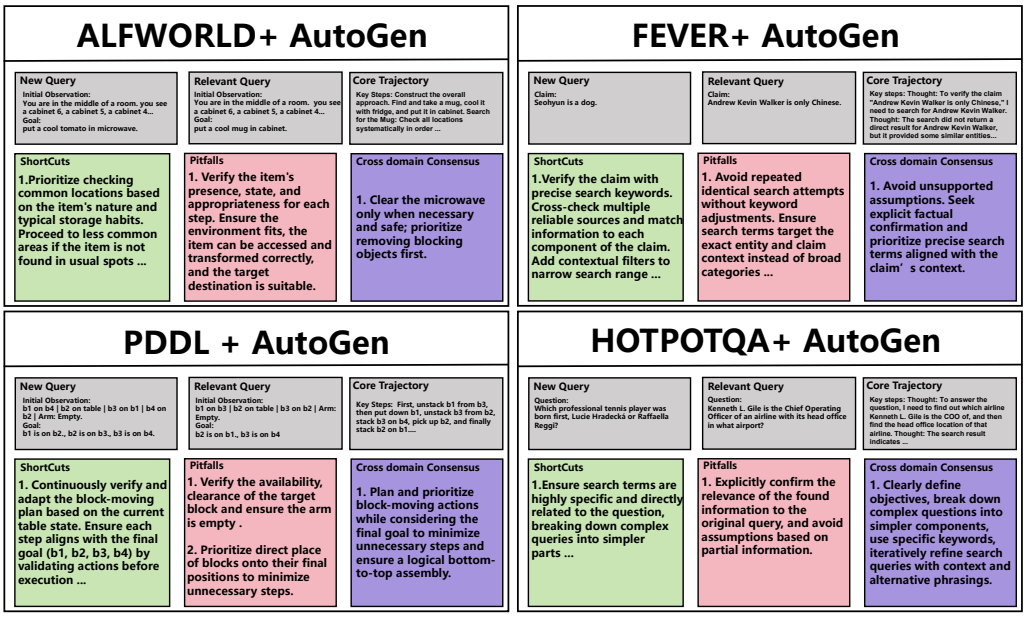

Figure 5: Case study of M-Memory.

A.2.2 Stability Evaluation

We summarize Round-3 performance using the mean and sample standard deviation across three independent repetitions of the complete three-round protocol. The statistical unit is an independent complete run, rather than an individual task or an individual round within a run.

We compare M-Memory with G-Memory using a one-sided Welch's t-test, retaining the directional hypothesis of improvement: $H_0 : \mu_M \leq \mu_G$ against $H_1 : \mu_M > \mu_G$. For sample means $\bar{x}_M, \bar{x}_G$, sample standard deviations s_M, s_G, and sample sizes $n_M = n_G = 3$, the test statistic and Welch–Satterthwaite degrees of freedom are

$$t = \frac{\bar{x}_M - \bar{x}_G}{\sqrt{s_M^2/n_M + s_G^2/n_G}},$$

$$\nu = \frac{\left(s_M^2/n_M + s_G^2/n_G\right)^2}{\frac{\left(s_M^2/n_M\right)^2}{n_M - 1} + \frac{\left(s_G^2/n_G\right)^2}{n_G - 1}}.$$

The one-sided p-value is $p_= \Pr(T_\nu \geq t)$.

For each backbone, we evaluate three MAS frameworks across five dataset–metric entries: ALFWorld, FEVER, HotpotQA, PDDL success rate, and PDDL reward rate. These correspond to four datasets and yield 15 comparisons per backbone. The significance threshold is set as $\alpha = 0.05$.

As shown in Tables 6 and 7, M-Memory improves significantly over G-Memory in 14 of 15 comparisons under Qwen-2.5-14b and 11 of 15 comparisons under gpt-4o-mini.

Table 6: One-sided Welch's t-tests comparing M-Memory with G-Memory under Qwen-2.5-14b at Round 3.

Dataset	AutoGen	DyLAN	MacNet
ALFWorld	$p =$ **0.0002458**	$p =$ **0.009860**	$p =$ **0.002192**
FEVER	$p =$ **0.000813**	$p =$ **0.002190**	$p = 0.051893$
HotpotQA	$p =$ **0.0034319**	$p =$ **0.0024188**	$p =$ **0.000706**
PDDL	$p =$ **0.009763**	$p =$ **0.003737**	$p =$ **0.018761**
PDDL(RE)	$p =$ **0.007663**	$p =$ **0.000271**	$p =$ **0.014144**

Table 7: One-sided Welch's t-tests comparing M-Memory with G-Memory under gpt-4o-mini at Round 3.

Dataset	AutoGen	DyLAN	MacNet
ALFWorld	$p =$ **0.035259**	**p=0.015045**	$p =$ **0.001659**
FEVER	$p = 0.500000$	$p =$ **0.000652**	$p =$ **0.018889**
HotpotQA	$p =$ **0.001245**	$p =$ **0.000520**	$p =$ **0.006625**
PDDL	$p = 0.500000$	$p = 0.313873$	$p =$ **0.000248**
PDDL(RE)	$p = 0.228102$	$p =$ **0.000697**	$p =$ **0.000387**

A.2.3 Held-out Evaluation

To further assess the generalization capability of the M-Memory, we conduct a held-out set evaluation. Specifically, each dataset is partitioned into a 60% memory construction set (serving as the training set) and a 40% held-out evaluation set (serving as the test set). This split allows us to examine how well the constructed memory generalizes to unseen data points.

As shown in Table 8, the best achievable performance on the 40% test set consistently falls short of the best performance obtained with the full dataset. The primary reason for this phenomenon lies in the relatively small scale of the benchmarks. On average, the four benchmarks contain only approximately 98.5 tasks each. Removing 40% of the data from the memory construction process substantially impairs the richness and diversity of the constructed memory, leading to a noticeable performance drop. This effect is particularly pronounced on more challenging benchmarks such as HotpotQA and PDDL, indicating that these tasks are more sensitive to the amount of available memory data.

In summary, the held-out evaluation confirms that while the M-Memory framework generalizes reasonably well to unseen data, its effectiveness is inherently constrained by the limited scale of existing benchmarks. This highlights the importance of developing more scalable memory construction strategies or collecting larger-scale evaluation benchmarks in future work.

A.3 Explanation

A.3.1 Update Rule

What a failure task evidences. A failure observed while $\iota_i \in \mathcal{I}_{\text{ret}}$ is consistent with at least five distinct causes: (C1) κ_i is misleading or wrong; (C2) κ_i is correct but the agents deviated from it during execution; (C3) κ_i is correct and was followed, but environmental stochasticity or intrinsic task hardness caused the failure; (C4) an incidental multi-agent error unrelated to κ_i (role confusion, communication conflict, collaboration deadlock); (C5) the attribution itself is wrong, *i.e.*, ι_i was retrieved but never actually used. Only C1 licenses weakening ι_i. A single-failure penalty therefore estimates $P(\text{C1} \mid \text{failure}, \iota_i \in \mathcal{I}_{\text{ret}})$ from a binary, uncalibrated signal that conflates C1 with C2–C5. Taking the more challenging PDDL tasks as an example, in the early rounds the success rate of task execution is extremely low, and the observed failures are dominated by the mixture of causes C2–C5; under such conditions, a direct penalty constitutes a severely biased estimator of insight quality.

Table 8: Held-out set evaluation of M-Memory and G-Memory across different MAS and datasets under Qwen-2.5-14b.

MAS	Dataset	M-Memory	G-Memory
AutoGen	ALFWorld	0.871	0.830
	FEVER	0.733	0.550
	HotpotQA	0.433	0.325
	PDDL	0.361	0.330
	PDDL (RE)	0.581	0.500
DyLAN	ALFWorld	0.852	0.852
	FEVER	0.700	0.575
	HotpotQA	0.350	0.317
	PDDL	0.292	0.278
	PDDL (RE)	0.619	0.439
MacNet	ALFWorld	0.852	0.815
	FEVER	0.710	0.600
	HotpotQA	0.417	0.367
	PDDL	0.333	0.306
	PDDL (RE)	0.631	0.536

Design rationale. Three arguments fix the conservative form of Eq. 12. **(R1) Evidence-type consistency.** Ω_i is a single-type statistic per perspective (success validation for $\mathcal{I}_S \cup \mathcal{I}_C$, failure exposure for $\mathcal{I}_P$); decrementing it on failure would mix "absence of validation" with "falsification" and destroy its interpretability. **(R2) Credit assignment and variance.** Penalizing every activated insight on every failure is an unattributed negative reward: it assigns the same update to causes C1–C5 and thus injects noise proportional to the failure rate rather than to insight error. **(R3) Bootstrap under sparse rewards.** A candidate insight needs survival time to acquire its first successful validation; on PDDL, where early-round success is rare, direct decay eliminates candidates before they can be validated at all.

Where failure information is written instead. Failure is not discarded; it is routed through five write paths, each with an explicit write, read-back, and behavioral consequence. **(W1) Pitfall distillation**: *write*—a new node in $\mathcal{I}_P$ whose support set is seeded by the failed query itself; *read-back*—Eq. 6 retrieves it whenever a future similarity neighborhood exhibits the recorded mode; *effect*—a warning enters Mem_i via Eq.8. **(W2) Hyper-edge binding**: *write*—the triple $(\iota_i, \iota_{\text{new}}, Q_{\text{new}})$ recording that ι_i was active when the failure occurred; *read-back*—hyperedge-aware traversal returns ι_{new} together with ι_i; *effect*—contextual caution, *i.e.*, functional weakening of ι_i without touching its statistic. **(W3) Rule-level auditing**: *write*—ADD/EDIT/REMOVE operations on the perspective rule sets driven by failed trials; *read-back*—the next retrieval of the perspective; *effect*—contradicted rules are rewritten or deleted by trial-based review. **(W4) Counterexample trajectories**: *write*—the failed trajectory is stored inside the query node; *read-back*—it may be selected as q_p^* and used by the sparsifier to filter error-prone actions; *effect*—negative behavioral guidance at the trajectory level.

When conservatism can hurt. The price of the above design is pruning delay: insights distilled from failed cases persist until either (i) a related insight contradicts them, in which case they are removed after a correctness judgment, or (ii) a sufficiently similar related insight emerges, in which case they are merged after a redundancy judgment. In practice, most such insights are ineffective merely because their summarization is insufficiently precise, and they progressively take effect as they are refined through successive merges. In certain cases, however, the delay is consequential: under non-stationary task streams, an insight κ_i that is valid in regime A but harmful in regime B is adapted by the framework only after failures have accumulated in B. Candidate mitigations, such as a periodic global audit every N tasks, are deferred to future work.

A.4 Concrete Illustration of Ablation Study

This section selects six cases for a case-by-case comparison in order to demonstrate the contribution of each dimensional perspective. The first three cases illustrate the effect of the Pitfall perspective, Cases 4 and 5 illustrate the effect of the Consensus perspective, and the sixth case runs in the opposite direction and serves to delineate the applicability boundary of the multi-dimensional insights.

A.4.1 Specificity of Search Queries

Case 1 is drawn from the FEVER dataset, with gpt-4o-mini as the backbone model, AutoGen as the MAS, and the task of determining the veracity of the statement "Janet Leigh was incapable of writing". Under the single-dimension configuration the task terminates with Finish[NOT ENOUGH INFO] and receives a reward of 0, whereas under the two-dimension configuration it correctly overturns the statement with Finish[REFUTES] and receives a reward of 1.

The insights injected by the two configurations exhibit a clear division of labour in their content. All of the insights provided by the single-dimension configuration revolve around the theme of multi-source cross-validation, for instance requiring the agent to "Ensure all claims are verified against accurate and reliable sources, including specific details and context, through thorough research and cross-referencing multiple authoritative sources", and further requiring a systematic evaluation of assertions together with a record of the retrieval process. These formulations are semantically correct, yet they remain normative appeals concerning verification behaviour and never touch upon how the retrieval behaviour itself should be constructed. The advice offered by the success dimension in the two-dimension configuration is highly similar to that of the single-dimension case, again concerning cross-validation and contextual analysis. The genuine difference originates from the failure dimension, which explicitly requires the agent to "Refine search queries to be specific and relevant, prioritize concise information gathering, and continuously verify the relevance and consistency of gathered evidence against credible sources", thereby shifting the optimisation target from whether the verification is sufficiently thorough to whether the search queries are sufficiently specific.

This difference is reflected directly in the trajectories. The solution process of the single-dimension configuration spans 12 steps and issues five searches in sequence, namely Search[Janet Leigh], Search[List of Janet Leigh performances], Search[Janet Leigh writing], Search[Janet Leigh writing works] and Search[Janet Leigh authored works]. All five queries are built around derivative forms of "writing" and are in essence synonymous retries within a single semantic field. After the fourth and the fifth search the model still judges that no information indicating Janet Leigh engaged in writing has been found, and it finally answers with insufficient evidence. The trajectory of the two-dimension configuration spans only 6 steps. After Search[Janet Leigh] fails to produce a direct hit, the model issues Search[Janet Leigh writing] as its second query, and the resulting observation states explicitly that Janet Leigh wrote four books, two of which are novels, on the basis of which the model immediately judges the original statement to be false. In other words, the constraint carried by the failure dimension pushes the search behaviour from synonymous retrying towards goal-directed retrieval, and what it saves is not merely steps but also one incorrect final judgement.

A.4.2 Ordering of Action Prerequisites

Case 2 is drawn from the tyreworld domain of PDDL, with gpt-4o-mini as the backbone model, AutoGen as the MAS, and the goal of completing a tyre change under the three conditions "Wheel r1 is inflated.", "r1 is on the-hub1." and "w1 is in boot". This task uses partial completion as its reward. The single-dimension configuration obtains 0.667, satisfying two of the three goals, while the two-dimension configuration obtains 1.0 with all three goals satisfied, and the environment returns "Wheel r1 is inflated. The goal is satisfied." at the final step.

The insight that plays the decisive role in the multi-dimensional configuration likewise comes from the failure dimension, and it reads "Maintain awareness of progress and ensure actions are performed in a logical order, addressing prerequisites before subsequent tasks to enhance efficiency and avoid invalid actions". It is worth noting that the success-dimension insight of the single-dimension configuration already contains a requirement of the form "Break tasks into sequential steps and validate each action", but what it emphasises is executing step by step and validating step by step, whereas what the failure dimension emphasises is that prerequisites should be satisfied before subsequent ac-

tions. “Stepwise” and “ordered” are two distinct constraints, the former constraining the granularity of execution and the latter constraining the dependency relations among actions.

The difference in the trajectories is precisely an embodiment of this distinction. The single-dimension configuration executes 60 steps and eventually exhausts its step budget. The order of its early actions is broadly reasonable, opening the boot, taking out the wrench and the jack, undoing nuts1, jacking up the-hub1, loosening and removing the old wheel w1, and placing w1 into the boot. At step 13, however, it executes put-away r1 in boot, thereby storing away in advance the new wheel r1 that should have been mounted onto the hub. This action destroys the precondition of the subsequent put-on-wheel r1, and the model is thereafter left oscillating among inflate r1, put-on-wheel r1, jack-down, do-up nuts1 and jack-up. The put-away r1 in boot issued at step 28 is judged by the environment as “The action is not valid and therefore takes no effect”, after which the trajectory degenerates into alternating outputs of check valid actions and think. The multi-dimensional configuration requires only 26 steps, following the order of loosening nuts1, jacking up the-hub1, loosening and removing w1, placing w1 into the boot, executing put-on-wheel r1 the-hub1, and finally inflate r1. The precondition of every step has been satisfied by a preceding step, and consequently no step is rejected by the environment. This comparison shows that the error recorded by the failure dimension concerning the ordering of prerequisites exactly hits the blind spot of the single-dimension configuration in long-horizon action planning.

A.4.3 Imbalance between Action and Reasoning

Case 3 is drawn from ALFWorld, with gpt-4o-mini as the backbone model and MacNet as the MAS, and the task is put a hot cup in cabinet. The single-dimension configuration fails to complete this task and the done flag is False, whereas the two-dimension configuration completes it successfully and the done flag is True.

The failure of the single-dimension configuration does not originate from an inability to locate the object or from a planning error, but from an imbalance in the output format. After localising the target cup the model emits the same thought several times in succession, “To solve the task, I need to find and take a cup, heat it with microwave, then put it in cabinet.”, and no longer issues any executable action. The trajectory is entirely repetitive between step 27 and step 30 and finally exhausts the budget. The trajectory of the two-dimension configuration spans 56 steps. At step 25 it executes heat cup 2 with microwave 1 and receives the feedback “You heat the cup 2 using the microwave 1.”, and at step 28 it executes put cup 2 in/on cabinet 1 and receives “You put the cup 2 in/on the cabinet 1.”, whereby the task is accomplished. In this case the failure-dimension insight of the multi-dimensional configuration requires the agent to “Prioritize actions based on task requirements and the environment, establishing a clear sequence to enhance clarity and efficiency while minimizing errors.”, and its effect can be understood as an explicit prompt to advance in an action-centric rather than a thought-centric manner.

A.4.4 Supplementing the Verdict Rule

Case 4 is drawn from the DyLAN configuration of FEVER, with qwen-2.5-14b as the backbone model, and the task is to judge the statement “Tool has won three Oscars.” The correct answer for this statement is REFUTES, since the retrieved results show clearly that the band won four Grammy Awards rather than any Oscar. The trajectory of the two-dimension configuration spans 24 steps, and the trajectory of the three-dimension configuration likewise spans 24 steps. The two are highly similar in their retrieval paths, both passing through Search[Tool], Search[Tool (band)] and the subsequent award-related searches, and both reading the same key observation, “Tool has won four Grammy Awards, but there is no mention of winning any Oscars”. The difference appears only in the verdict at the last step. The two-dimension configuration executes Finish[NOT ENOUGH INFO] and the environment returns “Answer is INCORRECT”, whereas the three-dimension configuration executes Finish[REFUTES] and the environment returns “Answer is CORRECT”.

The insights injected by the two-dimension configuration are dominated by retrieval-level requirements such as using precise search terms, cross-validating multiple sources and focusing on direct evidence. It should be stated faithfully that one of them does contain the phrase “support or refute”, yet its context is “checking for specific examples or instances that support or refute the claim”, which treats supporting or refuting as the objective of retrieval rather than prescribing when the agent

should answer NOT ENOUGH INFO and when it should answer REFUTES. The cross-domain perspective entry additionally injected by the three-dimension configuration states this verdict rule explicitly, "Break down claims into verifiable components ... refine search queries as needed, and explicitly state 'NOT ENOUGH INFO' if insufficient evidence is found, or conclude directly if the evidence clearly supports or refutes the claim." The reasoning at step 11 of the three-dimensional trajectory matches the wording of this rule closely, its exact phrase being "The claim states that Tool has won three Oscars, which is incorrect", meaning that the model first confirms that the evidence already obtained constitutes a negation of the claim and then settles on REFUTES. The attribution in this case is particularly strong because the two trajectories are entirely identical at the level of evidence, which leaves the availability of the verdict rule as the only variable.

A.4.5 SUPPLEMENTING THE LOCATION CHECKLIST

Case 5 is drawn from the DyLAN configuration of ALFWorld, with qwen-2.5-14b as the backbone model, and the task is put some saltshaker on cabinet. The two-dimension configuration exhausts its entire step budget and the done flag is False, whereas the three-dimension configuration completes the task after 21 steps and the done flag is True. The two trajectories begin almost identically. In the thought at the second step both models enumerate a list of candidate locations, "A saltshaker is more likely to appear in cabinet (1-6), countertop (1-3), drawer (1-3)...", and both then examine the candidates one by one starting from cabinet 1.

The cross-domain perspective entry additionally injected by the three-dimension configuration provides a more concrete location checklist, "Systematically organize countertops, drawers, cabinets, shelves, and areas behind appliances, ensuring all items are properly placed and drawers are fully accessible." Among these items, "areas behind appliances" is a location that is easily skipped in the ALFWorld environment, and it therefore brings a more complete search coverage.

A.4.6 THE APPLICABILITY BOUNDARY

The manner in which the cross-domain perspective dimension is constructed determines that it necessarily has a failing side. This dimension is retrieved and abstracted by the graph memory from a set of diverse successful tasks. Consequently, when the retrieved source task is dissimilar to the current task yet still shares surface vocabulary with it, the abstracted entry is injected in the form of out-of-domain knowledge. We observe such an entry in some successful runs, "Ensure the hub is jacked up before attempting to undo the fastening of the nut, and check valid actions after each significant step, because this prevents damage to the wheel and ensures the task can proceed smoothly while avoiding invalid actions."

This piece of advice itself originates from the tyreworld subdomain of PDDL, and it is specific in wording and highly actionable. The task into which it is injected, however, has an initial observation of "B1 is on b3. B2 is on the table. B3 is on b2. B4 is on b5. B5 is on b1. B6 is on b4.", which belongs to the blocksworld subdomain, where there is neither a hub nor a nut nor a jack, so the entire piece of advice is completely disjoint from the action space of the current task. This task does succeed under the three-dimension configuration, but the success cannot be attributed to this advice. The phenomenon suggests that the positive gain of the cross-domain perspective dimension at certain moments does not come from the formal property of having more dimensions, but from a small number of entries that happen to carry transferable verdict rules or checklists. Most of the remaining entries are generic formulations that can be attached to any task, and a few, as in the example above, constitute cross-domain mismatched noise whose cost is borne by prompt length and attention dilution.

A.5 PROMPT SET

A.5.1 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.
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:

A.5.2 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:

A.6 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.

A.6.1 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 diverse successful tasks. You may consider whether you can directly apply 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}

A.6.2 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:

A.7 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).