MeMAD: Structured Memory of Debates for Enhanced Multi-Agent Reasoning

Shuai Ling[1,3*] **Lizi Liao**[2] **Dongmei Jiang**[3†] **Weili Guan**[1†]

[1]Harbin Institute of Technology (Shenzhen)
[2]Singapore Management University
[3]Pengcheng Laboratory

24b952037@stu.hit.edu.cn lzliao@smu.edu.sg
jiangdm@pcl.ac.cn honeyguan@gmail.com

Abstract

Large Language Models (LLMs) demonstrate remarkable in-context learning capabilities but often struggle with complex, multi-step reasoning. Multi-Agent Debate (MAD) frameworks partially address these limitations by enabling iterative agent interactions. However, they neglect valuable historical insights by treating each new debate independently. In this paper, we propose Memory-Augmented MAD (MeMAD), a parameter-free memory-augmented MAD framework that systematically organizes and reuses past debate transcripts. MeMAD stores structured representations of successful and unsuccessful reasoning attempts enriched with self-reflections and peer feedback. It systematically retrieves them via semantic similarity at inference time to inform new reasoning tasks. Our experiments on challenging mathematical reasoning, scientific question answering, and language understanding benchmarks show that MeMAD achieves significant accuracy gains (up to 3.3% over conventional MAD baselines) without parameter updates. Our findings underscore structured memory as a pivotal mechanism for achieving deeper and more reliable multi-agent reasoning in LLMs. Code is available in https://github.com/LSHCoding/MeMAD.

1 Introduction

LLMs have significantly advanced the field of natural language processing, exhibiting remarkable in-context learning capabilities (Brown et al., 2020; OpenAI, 2024; Zhao et al., 2024; Wei et al., 2022). However, despite these successes, LLMs often encounter difficulties when dealing with complex reasoning tasks that necessitate multi-step reasoning. Many approaches to enhance reasoning capabilities typically rely on parameter updates through methods such as fine-tuning, but these methods demand extensive high-quality data, substantial computational resources and are susceptible to catastrophic forgetting. On the other hand, the impressive in-context learning ability of LLMs has motivated a parameter-free learning paradigm, enabling adaptation to diverse tasks through prompt engineering rather than parameter modifications (Min et al., 2022; Dong et al., 2024).

Under this parameter-free paradigm, existing research primarily explores two streams: *Single-agent methods*, employ specialized prompting techniques (Zhou et al., 2023; Besta et al., 2024)—such as chain-of-thought reasoning (Wei et al., 2023), multi-path sampling (Wang et al., 2023), and iterative self-refinement (Madaan et al., 2024; Shinn et al., 2023)—to improve reasoning performance. Alternatively, *Multi-agent methods* simulate human-like collaborative problem-solving by facilitating iterative interactions among multiple agents, thereby enabling mutual correction and knowledge exchange (Chan et al., 2023; Hong

*Work done during an internship in SMU.
†Corresponding authors

et al., 2024; Chen et al., 2024b). Among these, Multi-agent Debate (MAD) is an effective representative (Du et al., 2023; Li et al., 2024). Although MAD has demonstrated promise in enhancing reasoning, current implementations typically treat each debate independently, discarding valuable debate transcripts once an answer is finalized. This practice neglects the rich reflections and iterative corrections embedded in historical interactions. It severely limits the system's capacity for continuous improvement, especially in real-world scenarios where similar problems repeatedly arise under slightly varied contexts.

Recently, memory mechanisms have emerged as powerful tools for enhancing LLMs' abilities in long-term knowledge retention and dynamic adaptation (Modarressi et al., 2025; Zhong et al., 2024; Packer et al., 2024; Wang et al., 2024a). It signals a viable way to further improve multi-agent reasoning. For example, Xu et al. (2025) proposed an agentic memory system that significantly improves long-term interaction capabilities and performance on complex tasks such as multi-hop reasoning. Despite their effectiveness, existing memory-augmented approaches predominantly emphasize general memory operations (e.g., reading, writing, and retrieval) and overcome the limitations of context window size. However, they often neglect transforming historical debate processes, especially valuable agent interactions, into structured, reusable experiences. This oversight constitutes a critical gap that hinders the systematic accumulation and reuse of valuable reasoning insights within MAD frameworks. Consequently, it motivates targeted exploration of memory mechanisms specifically tailored for multi-agent debates.

To bridge this critical gap, we propose **Me**mory-Augmented **MAD** (named as **MeMAD**), a parameter-free framework that organizes and reuses debate experiences to guide future multi-agent interactions as illustrated in Figure 1. MeMAD stores evidence of successful and failed reasoning pathways—complete with self- and peer-feedback—in a structured memory that prioritizes both the context of each problem and the insights gleaned from the debate process. During inference, it retrieves and injects relevant experiences into agent prompts via semantic matching, ensuring that past lessons directly inform current debates without requiring any parameter updates. This design allows for continuous knowledge accumulation, enabling LLMs to learn from and build upon collective historical experiences. We evaluate MeMAD on challenging tasks across diverse domains and our experiments demonstrate that MeMAD significantly outperforms baselines without requiring parameter updates.

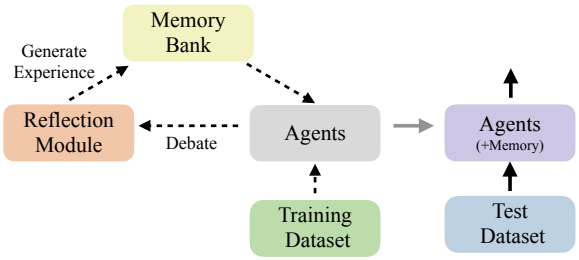

Figure 1: Memory enhancement without parameter update.

In summary, our contributions are threefold: 1) We propose a parameter-free framework that systematically stores and reuses past multi-agent debate experiences to enhance complex reasoning. 2) Our approach enriches debate records with self-reflection and peer-generated insights, capturing both successful and failed reasoning pathways for better usage. 3) Experiments on diverse complex reasoning tasks show consistent and significant accuracy improvements, demonstrating the value of reusing debate experiences.

2 Related Work

2.1 Multi-Agent Debate

The Multi-Agent Debate (MAD) framework leverages collaborative interactions among agents to enhance reasoning, inspired by the "society of minds" concept. Building on the foundational work of (Du et al., 2023), recent studies have explored two key directions to improve MAD: differentiated agents and mechanism refinement. Differentiated-agent methods either assign distinct roles and expertise to agents (Chan et al., 2023; Liang et al., 2023; Li et al., 2023) or employ diverse pre-trained models to broaden the scope of debates (Chen

et al., 2024b; Wang et al., 2024c). Mechanism refinement focuses on optimizing debate flow and reducing redundancy to improve efficiency (Liu et al., 2024; Li et al., 2024; Sun et al., 2024). However, existing MAD frameworks generally treat each debate independently, overlooking the structured accumulation and reuse of historical debate experiences. In contrast, our method transforms the debate process into reusable parameter-free experiences, enabling cross-task improvements that traditional MAD frameworks have yet to achieve.

2.2 Memory Augmented LLMs

Recent works on memory augmentation for LLMs have explored various mechanisms to enhance long-term knowledge retention and dynamic memory updating (Modarressi et al., 2025; Zhong et al., 2024; Packer et al., 2024; Wang et al., 2024a). Some approaches employ explicit read-write memory architectures or parameter fine-tuning techniques to permanently integrate new knowledge (Modarressi et al., 2025; Wang et al., 2024d; Zhong et al., 2024), but these methods often incur high computational costs. Alternatively, non-parametric methods leverage external knowledge retrieval (Qian et al., 2024) or compression-based storage strategies (Wang et al., 2025; 2024a) to dynamically update and efficiently retrieve relevant information. Additionally, cognitive-inspired frameworks introduce selective forgetting and memory reinforcement mechanisms (Zhong et al., 2024) to further enhance long-term interaction capabilities. Nonetheless, existing memory-augmentation methods predominantly focus on general memory operations (reading, writing, retrieval) and overcoming the limitations of context window size, largely neglecting the structured transformation of multi-agent debate interactions into reusable reasoning experiences. Our MeMAD framework addresses this limitation by systematically structuring and reusing debate transcripts enriched with reflective insights, thereby significantly enhancing multi-agent reasoning without parameter updates.

3 Problem Setup

To systematically describe the process of MAD and establish a unified notation system for subsequent methodological discussions, this section formalizes the core elements of the MAD framework. Consider a set of N agents $\mathcal{A} = \{A_1, A_2, \ldots, A_N\}$ engaged in a debate about a given question Q. The debate process consists of T iterative rounds. At each round $t \in \{1, 2, \ldots, T\}$, each agent A_i generates a response $O_{i,t}$ based on the question Q and the debate history up to the previous round, denoted as $\mathcal{H}_{i,t-1}$ (where the initial state is $\mathcal{H}_{i,0} = \emptyset$). The debate history is subsequently updated as $\mathcal{H}_{i,t} = \{\mathcal{H}_{i,t-1}, O_{1,t}, \ldots, O_{N,t}\}$, which serves as the contextual information for the next round for agent A_i.

If a *feedback mechanism* is incorporated, agents provide both *self-feedback* and *peer-feedback* at the end of each round. Specifically, after all agents produce their responses $\{O_{1,t}, \ldots, O_{N,t}\}$ in round t, each agent A_i generates self-feedback $SF_{i,t}$ for its own response and peer-feedback $PF_{i,j,t}$ for the response of agent A_j. These feedback signals are then appended to the debate history, updating it to $\mathcal{H}_{i,t} = \{\mathcal{H}_{i,t-1}, O_{1,t}, \ldots, O_{N,t}, SF_{i,t}, \{PF_{j,i,t}\}_{j=1}^{N}\}$ and enriching the context for the next round of debate. After T rounds, all agents participate in a voting process to determine the final answer, resulting in the debate outcome R_{debate}.

4 The MeMAD Method

We propose Memory-Augmented Multi-Agent Debate (MeMAD) to overcome the challenge of underutilized experiences in traditional MAD systems. As depicted in Figure 2, MeMAD integrates two key components—*experience accumulation* and *memory retrieval*—within the debate loop. In the experience accumulation phase, each debate is augmented with both self- and peer-feedback, which are systematically stored in a structured memory to capture reasoning trajectories and reflective insights. In the inference phase, the memory retrieval mechanism leverages these structured records to guide ongoing debates, facilitating more informed and efficient multi-agent reasoning. Section 4.1 delves into the dual-level feedback

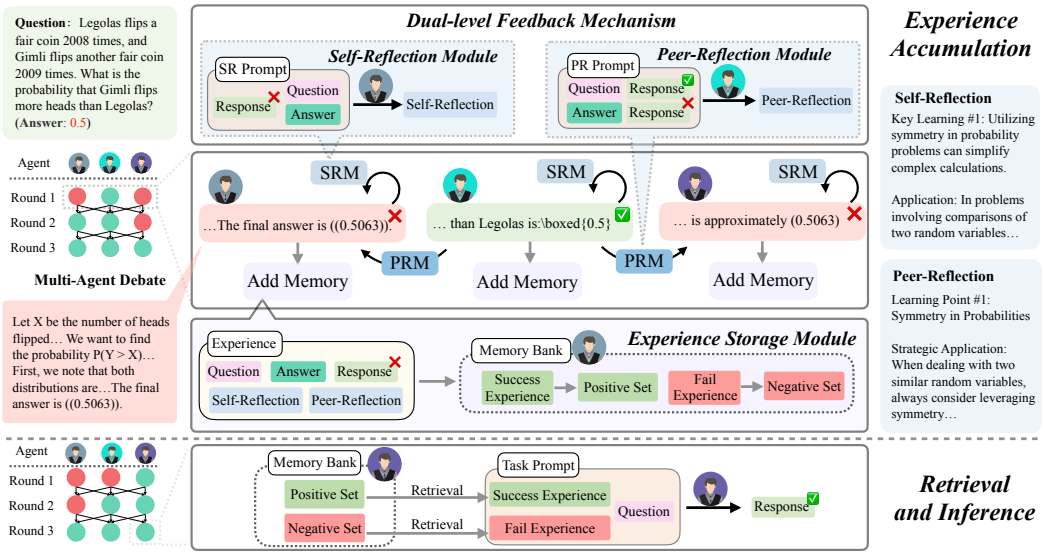

Figure 2: Overview of the MeMAD, consisting of two components: (1) *Experience Accumulation* (top), featuring a *Dual-level Feedback Mechanism* with *Self-Reflection* and *Peer-Reflection* Modules, followed by an *Experience Storage Module* that categorizes reasoning attempts as positive or negative examples; and (2) *Retrieval and Inference* (bottom), where relevant past experiences are retrieved from the Memory Bank to inform new debates.

and structured storage process, while Section 4.2 details how these accumulated experiences are retrieved and applied to enhance performance across diverse reasoning tasks.

4.1 Experience Accumulation: Construction of Experience-driven Memory Bank

The experience accumulation phase is a pivotal component of the MeMAD framework. As depicted in Figure 2, unlike traditional MAD frameworks, MeMAD introduces an innovative dual-level feedback mechanism and a structured experience storage system during this phase. The primary objective of this phase is to construct a memory bank comprising rich, high-quality, and well-structured debate experiences, thereby laying a solid foundation for subsequent experience retrieval and reasoning. To achieve this goal, MeMAD employs a dual-level feedback mechanism, which enables the multi-agent system to generate high-quality, reusable structured experiences from two complementary dimensions: self-reflection (Self-Feedback) and mutual assistance (Peer-Feedback).

4.1.1 Dual-level Feedback Mechanism

To facilitate effective learning and experience accumulation during the debate process, the MeMAD framework employs a dual-level feedback mechanism. This mechanism evaluates and provides feedback on each agent's debate performance at the end of every debate round by leveraging the ground truth answer to the question as an external supervision signal. Specifically, at the conclusion of each debate round t, the ground truth answer is used to assess the correctness of each agent A_i's response $O_{i,t}$, generating a binary supervision signal $C_{i,t} \in \{\text{True}, \text{False}\}$. Building upon this binary supervision signal and the ground truth answer, the MeMAD framework incorporates self-feedback and peer-feedback mechanisms, which enable the generation of agent experiences from two complementary perspectives: intrinsic self-reflection and external peer evaluation.

Self-Feedback. Each agent A_i engages in a self-feedback process upon receiving the supervision signal $C_{i,t}$, which involves a thorough reflection on its reasoning mechanism. The self-feedback, denoted as $SF_{i,t}$, is a textual output generated by agent A_i based on its historical context $\mathcal{H}_{i,t-1}$ and the supervision signal, guided by a predefined self-feedback

prompt template P_{sf} (see Appendix E.1 for details). The primary objective of this process is to analyze and identify the underlying factors contributing to the successes or failures observed during the current round of debate. This analysis facilitates the formation of experiential knowledge that can be generalized to future scenarios. The formal definition of $SF_{i,t}$ is given as:

$$SF_{i,t} = A_i\left(P_{sf}(Q, Y, O_{i,t}, C_{i,t}), \mathcal{H}_{i,t-1}\right).$$

Here, $P_{sf}(Q, Y, O_{i,t}, C_{i,t})$ is the self-feedback prompt constructed based on the question Q, the ground truth answer Y, the agent output $O_{i,t}$, and the correctness signal $C_{i,t}$.

Peer-Feedback. To effectively facilitate collaborative learning and knowledge sharing among agents, the MeMAD framework incorporates a peer-feedback mechanism. When agent A_j provides a correct answer in a debate while agent A_i answers incorrectly, A_j is responsible for offering peer feedback to A_i. This feedback involves identifying deficiencies in A_i's response and providing actionable suggestions for improvement.

The peer feedback, denoted as $PF_{j,i,t}$, is generated by the correct-answering agent A_j based on its historical context $\mathcal{H}_{j,t-1}$ and supervisory information. The generation process is guided by a predefined peer-feedback prompt template P_{pf} (see Appendix E.1 for details), which takes as input the question Q, the ground-truth answer Y, the output $O_{i,t}$ of agent A_i, and the output $O_{j,t}$ of agent A_j. Formally, the peer feedback is defined as:

$$PF_{j,i,t} = A_j\left(P_{pf}(Q, Y, O_{i,t}, O_{j,t}), \mathcal{H}_{j,t-1}\right).$$

4.1.2 *Experience Storage Module*

At the end of each debate round t, the key information generated by each agent A_i is organized into an experience tuple $E_{i,t}$, which is subsequently stored in the agent's dedicated memory bank $\mathcal{M}_{i,t}$. The memory bank is composed of a positive set $\mathcal{M}^{+}_{i,t}$ and a negative set $\mathcal{M}^{-}_{i,t}$. The specific storage location is determined by the agent's response: successful experiences are stored in $\mathcal{M}^{+}_{i,t}$, while failed experiences are stored in $\mathcal{M}^{-}_{i,t}$. The structure of the experience tuple $E_{i,t}$ is formally defined as:

$$E_{i,t} =< Q, Y, O_{i,t}, Ref_{i,t} > .$$

Here, $Ref_{i,t}$ refers to the reflective experience, which integrates both self-feedback and peer-feedback, aiming to provide effective guidance for future debates. Specifically, the construction of $Ref_{i,t}$ is defined as:

$$Ref_{i,t} = \begin{cases} SF_{i,t}, & \text{if } C_{i,t} = \text{True}, \\ SF_{i,t} \cup \{PF_{j,i,t}\}_{j=1}^{N}, & \text{if } C_{i,t} = \text{False}. \end{cases}$$

The memory bank $\mathcal{M}_{i,t}$ serves as a dedicated storage space for each agent, systematically accumulating all structured experiences generated during the experience accumulation phase. As the number of debate rounds increases, $\mathcal{M}_{i,t}$ is continuously expanded, laying a solid foundation for efficient experience retrieval and utilization during the testing phase.

4.2 Retrieval and Inference

After the experience accumulation phase, the memory bank is utilized during the multi-agent debate process. As shown in Figure 2, a key innovation of MeMAD compared to traditional MAD is its integration of structured memory. At the start of each debate round, agents retrieve relevant historical experiences from their dedicated memory banks based on the current question. These retrieved experiences are then injected into the prompt, enabling agents to leverage past knowledge to enhance reasoning capabilities and debate efficiency. The memory bank is implemented using Chroma, an open-source AI application database.

During the testing phase, at the beginning of each debate round t, for a given test question Q^{test}, each agent A_i retrieves relevant experiences from its memory bank $\mathcal{M}_{i,t}$. The MeMAD framework employs a semantic similarity-based retrieval mechanism to identify pertinent experiences. The detailed steps are as follows:

1. The test question Q^{test} and the agent's historical answer $O_{i,t-1}$ (empty in the initial round) are semantically encoded into a query vector $\mathbf{q}_{i,t}$ using a Sentence-BERT encoder (Chen et al., 2024a):
$$\mathbf{q}_{i,t} = \text{Encoder}([Q^{test}; O_{i,t-1}]).$$
2. For each experience $E^k_{i,t} = \langle Q^k, Y^k, O^k_{i,t}, Ref^k_{i,t} \rangle$ in the memory bank $\mathcal{M}_{i,t} = \mathcal{M}^+_{i,t} \cup \mathcal{M}^-_{i,t}$, its corresponding question Q^k and answer $O^k_{i,t-1}$ are encoded into a vector $\mathbf{v}^k_{i,t}$. The cosine similarity between $\mathbf{q}_{i,t}$ and $\mathbf{v}^k_{i,t}$ is then computed.
3. With similarity scores, the top-K experiences are selected from both the positive memory set $\mathcal{M}^+_{i,t}$ and the negative memory set $\mathcal{M}^-_{i,t}$, forming the retrieved set:
$$\mathcal{E}_{retrieved} = \{E^{1,+}_{i,t}, ..., E^{K,+}_{i,t}, E^{1,-}_{i,t}, ..., E^{K,-}_{i,t}\}.$$

Once the relevant experiences $\mathcal{E}_{retrieved}$ are retrieved, they are injected into the agent's prompt to guide the debate process:

$$O_{i,t} = A_i \left(P_{task}(Q^{test}, \mathcal{E}_{retrieved}), \mathcal{H}_{i,t-1} \right).$$

By leveraging retrieved experiences, the MeMAD framework enables agents to avoid repeating past mistakes and adopt more effective reasoning strategies. This significantly improves both the efficiency and accuracy of the debate process during the testing phase.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate our method across four datasets spanning mathematics, science, law, and economics. Specifically, we use (1) MATH500 (a subset of the MATH benchmark with advanced math problems) (Lightman et al., 2023), (2) GPQA (graduate-level multiple-choice questions in biology, physics, and chemistry) (Rein et al., 2023), and (3) two specialized subsets of MMLUPro (we denote as Law and Economics correspondingly) (Wang et al., 2024e). Each dataset is split into training and test sets to ensure sufficient coverage and diversity. Detailed descriptions and selection criteria are provided in the Appendix B.

Baselines. The baselines for comparison can be categorized into three main groups:

- **Vanilla single-agent methods**. We employed OpenAI's GPT-4o-mini, ChatGPT, and the open-source Qwen2.5-14b model (Yang et al., 2024). All tasks were performed using the same task prompts as those adopted in the MeMAD benchmark.
- **Advanced single-agent methods**. Such methods go beyond a single pass of generation by employing more nuanced mechanisms for feedback and exploration. (1) Self-Refine iteratively improves its output by generating internal feedback and refining its own responses (Madaan et al., 2024), while (2) Self-Consistency samples multiple reasoning paths and aggregates the final answers to mitigate the impact of any single flawed chain of thought (Wang et al., 2023). (3) Reflexion complements these approaches by reinforcing decision-making through linguistic feedback, enabling agents to reflect on and learn from their own outputs (Shinn et al., 2023).
- **Multi-agent methods**. We utilized two multi-agent-based methods as baselines: Multi-Agent Debate (MAD) (Du et al., 2023) and Mixture-of-Agents (MoA) (Wang et al., 2024b).

Implementation Details. Considering the inherent cost associated with API-based models and in line with established practices in prior research, we adopted a dataset sampling strategy in this study Chen et al. (2024b). For the Law and Economics datasets, we randomly sampled a portion of the data as the test set. For MATH500, we selected the most challenging problems as the test set (with a level of 5). For GPQA, we used the GPQA Diamond subset as the test set. All agents were built using the GPT-4o-mini model provided by OpenAI.

5.2 Main Results

Category	Method	MATH500	GPQA	Law	Economics	Average
Vanilla single-agent	GPT-4o mini	0.485	0.379	0.395	0.683	0.486
	ChatGPT	0.179	0.281	0.342	0.550	0.338
	Qwen2.5-14B	0.500	0.429	0.370	0.673	0.493
Advanced single-agent	Self-Refine	0.537 (+0.052)	0.438 (+0.059)	0.383 (-0.012)	0.692 (+0.009)	0.513 (+0.027)
	Self-Consistency	0.560 (+0.075)	0.402 (+0.023)	0.411 (+0.016)	0.673 (-0.010)	0.512 (+0.026)
	Reflexion	0.560 (+0.075)	0.414 (+0.035)	0.412 (+0.017)	0.713 (+0.030)	0.525 (+0.039)
Multi-agent	MAD	0.552 (+0.067)	0.409 (+0.030)	0.425 (+0.030)	0.701 (+0.018)	0.522 (+0.036)
	MoA	0.537 (+0.052)	0.419 (+0.040)	0.435 (+0.040)	0.763 (+0.080)	0.539 (+0.053)
	MeMAD (ours)	**0.590 (+0.105)**	**0.460 (+0.081)**	**0.457 (+0.062)**	**0.782 (+0.099)**	**0.572 (+0.086)**
	Improvement	↑ 0.030	↑ 0.022	↑ 0.022	↑ 0.019	↑ 0.033

Table 1: Performance comparison of different methods across datasets. Green text indicates improvements over GPT-4o-mini, gray text signifies declines compared to GPT-4o-mini, and red text highlights MeMAD's advancements over other top-performing methods.

MeMAD consistently outperforms all baseline methods. Table 1 presents comprehensive experimental results across multiple reasoning tasks. Overall, MeMAD achieves SOTA performance, surpassing the strongest baseline, MoA, by 3.3% and improving upon the MAD framework by 5%. These results highlight the effectiveness of structured memory augmentation in multi-agent debates. Specifically, MeMAD achieves performance gains of 3.0%, 2.2%, 2.2%, and 1.9% on the MATH500, GPQA, Law, and Economics datasets, respectively. This consistent improvement across domains demonstrates the generalizability.

Advanced single-agent methods show incremental but limited improvements. Within the Advanced single-agent category, Reflexion achieves the best overall performance, outperforming Self-Refine and Self-Consistency on three out of four datasets. Reflexion's advantage lies in its memory mechanism, which enables iterative reasoning and feedback reuse, allowing it to capture and utilize historical insights. In contrast, Self-Refine relies on iterative refinement strategies but lacks the memory-driven enhancements of Reflexion, limiting its effectiveness. Self-Consistency, which employs multi-path sampling akin to a simplified multi-agent debate, demonstrates modest gains but fails to fully leverage collaborative reasoning. These results suggest that while advanced single-agent methods improve upon vanilla approaches, their lack of structured memory integration constrains their ability to tackle complex reasoning tasks.

Multi-agent methods demonstrate superior performance. The Multi-agent methods consistently outperform single-agent approaches, with MeMAD achieving the highest accuracy across all evaluation tasks. The significant improvements over MAD validate our core hypothesis that structured experience reuse enhances multi-agent reasoning. Importantly, our parameter-free framework for experience accumulation and retrieval can be readily integrated with other multi-agent architectures, offering a promising direction for future research in collaborative reasoning systems.

5.3 Detailed Analysis and Ablations of MeMAD

Comprehensive Memory Components Enhance Reasoning Performance. To systematically evaluate the impact of different memory components on MeMAD's performance, we conducted an ablation study on the Economics dataset. Starting from a minimal memory configuration containing only questions and solutions (Q + Y), we incrementally incorpo-

rated additional components: agent responses (O), self-reflection (SR), and peer-reflection (PR). The results, as shown in Table 2, clearly demonstrate the cumulative benefits of these components.

The baseline configuration (Q + Y) achieved an accuracy of 75.4%, providing a foundational understanding of the problem and its solution. Adding agent responses and self-reflection (Q + Y + O + SR) led to a 1.4% improvement, reaching 76.8%. This highlights the value of incorporating iterative reasoning and self-assessment into the memory structure. The full configuration, which integrates both self- and peer-reflections (Q + Y + O + (SR ∪ PR)), achieved a further improvement, reaching 77.7%. These results validate that optimal performance requires both complete contextual information and multi-perspective feedback, aligning with our framework design in Section 4.

Memory	Acc
∅	0.701
Q + Y	0.754 (+0.053)
Q + Y + O + SR	0.768 (+0.067)
Q + Y + O + (SR ∪ PR)	**0.777 (+0.076)**

Table 2: The impact of different experience compositions on performance: question (Q), solution (Y), agent response (O), self-reflection (SR), and peer-reflection (PR).

Embedding-based retrieval surpasses token-overlap-based retrieval. To evaluate the effectiveness of different retrieval methods within MeMAD (Li & Li, 2024; Lee et al., 2024), we conducted a comparative analysis of token-overlap-based and embedding-based approaches, as summarized in Table 3. Overall, embedding-based approaches outperform the traditional token-overlap-based BM25 method, underscoring the importance of semantic understanding in retrieving debate experiences.

Among the embedding-based methods, nomic-embed-text achieved the best performance on the Economics (0.782) and Law (0.457) datasets, showcasing its strong ability to handle domain-specific tasks that require nuanced semantic understanding. On the other hand, bge-m3 excelled on the MATH500 (0.590) and GPQA (0.460) datasets, indicating its robustness in mathematical reasoning and scientific question answering tasks. Notably, bge-m3 emerged as the best-performing method overall, with an average accuracy of 0.567, outperforming nomic-embed-text (0.553) and mxbai-embed-large (0.548). Additionally, BM25 performs comparably to bge-m3 on MATH500 (0.590), suggesting that the mathematical reasoning dataset may rely more on lexical matching than on deep semantic representations.

Category	Method	MATH500	Economics	GPQA	Law	Average
Token-overlap	BM25	0.590	0.739	0.419	0.424	0.543
Embedding	bge-m3	**0.590 (+0.00)**	0.777 (+0.038)	**0.460 (+0.041)**	0.440 (+0.016)	**0.567 (+0.024)**
	nomic-embed-text	0.560 (-0.03)	**0.782 (+0.043)**	0.414 (-0.005)	**0.457 (+0.033)**	0.553 (+0.010)
	mxbai-embed-large	0.582 (-0.008)	0.744 (+0.005)	0.455 (+0.036)	0.409 (-0.015)	0.548 (+0.005)

Table 3: Impact of different embedding methods. Green text indicates improvements over BM25 and gray text signifies declines compared to BM25.

MeMAD outperforms other experience selection strategies. To systematically evaluate the impact of different experience selection strategies and memory configurations on MeMAD's performance, we conducted comprehensive ablation studies on the Economics dataset. We compared three experience selection strategies: (1) *Random*, which selects N cases randomly from the memory bank; (2) *Similarity*, which retrieves the top-N cases with the highest semantic similarity to the current problem; and (3) *Diversity*, which first clusters the experiences using K-Means and then selects one semantically similar case per cluster to ensure diversity. To ensure fairness, all experiments were constrained to select the same number of experiences. Additionally, we examined the contributions of successful ($\mathcal{M}^+$) and failed ($\mathcal{M}^-$) debate experiences, as well as their combination ($\mathcal{M}^+ \cup \mathcal{M}^-$), to assess the role of memory bank composition. Table 4 summarizes the results, which clearly demonstrate that memory augmentation consistently enhances reasoning accuracy across all methods, validating the importance of leveraging historical debate experiences.

Selection Strategy	$\mathcal{M}^+$	$\mathcal{M}^-$	$\mathcal{M}^+ \cup \mathcal{M}^-$
Random	0.730 (+0.029)	0.720 (+0.019)	0.739 (+0.038)
Similarity	0.735 (+0.034)	0.739 (+0.038)	0.754 (+0.053)
Diversity	0.739 (+0.038)	0.754 (+0.053)	0.768 (+0.067)
Positive and Negative	-	-	**0.777 (+0.076)**

Table 4: Performance of different experience selection strategies on Economics. We use *bge-m3* as it performed the best across datasets on average.

The results in Table 4 highlight several key findings. First, memory augmentation consistently improves performance over the MAD across all selection strategies, affirming the utility of historical debate experiences (Green colored numbers indicate the improvements over MAD). However, the choice of experience selection strategy significantly influences effectiveness. *Random* yields only marginal gains (2.9%), while *Similarity* retrieval achieves substantial accuracy improvements (4.2%). Specifically, *Similarity* retrieval outperforms *Random* by 1.3% on average, demonstrating the importance of selecting contextually relevant experiences. Second, we observe that *Diversity*, which balances semantic similarity with diversity, achieves better performance. Finally, our analysis reveals that our proposed *Positive and Negative* strategy, adding in both successful and failed debate experiences, yields the highest performance (0.777). Notably, methods over the negative set $\mathcal{M}^-$ sometimes perform comparably to or even outperform methods over the positive set $\mathcal{M}^+$, suggesting that learning from failed reasoning attempts provides unique insights. Methods over the combined ($\mathcal{M}^+ \cup \mathcal{M}^-$) leverage the complementary strengths of both sets, achieving optimal results. These findings validate MeMAD's core design principle of learning from both successes and failures, underscoring the importance of a comprehensive and structured memory.

5.4 Transferability and Generality

MeMAD demonstrates cross-task knowledge transferability. To evaluate the cross-task experience transferability of the MeMAD framework, we conducted an experience transfer experiment, as summarized in Table 5. In this experiment, we utilized experiences accumulated from the MATH500 task to improve performance on the MMLUPro-Math task, which differs significantly in task format. To minimize the impact of format-specific features, the experience repository was curated to retain only task-format-independent content, specifically self-reflective and peer-reflective feedback. As shown in Table 5, MeMAD achieves a substantial accuracy improvement of 4.0% on the MMLUPro-Math task (from 0.725 to 0.765) compared to the MAD baseline. This result strongly validates the cross-task generalization capabilities of the MeMAD framework, even when these tasks differ significantly in structure. Furthermore, this experiment indirectly corroborates the effectiveness of the dual-level feedback mechanism, which enriches the memory bank with transferable insights.

Method	Acc
MAD	0.725
MeMAD	**0.765**

Table 5: Evaluation of experience transfer performance.

MeMAD demonstrates generality across stronger LLMs. To evaluate the generality of the MeMAD framework, we designed a generality experiment, with the results summarized in the table 6. Specifically, we conducted experiments on the GPQA and Economics datasets, leveraging the experience accumulated by GPT-4o-mini to enhance strongger models, GPT-4o and Deepseek-V3. Compared to MAD, MeMAD exhibited performance improvements across all models on both datasets, highlighting the generality of our approach to more advanced models. Additionally, we observed that even experience derived from weaker models can be effectively transferred to enhance the performance of stronger models.

6 Conclusion

In this paper, we introduced Memory-Augmented Multi-Agent Debate (MeMAD), a novel parameter-free framework that systematically captures and reuses debate experiences to enhance complex reasoning. By integrating a structured memory system with dual-level feedback—comprising both self-feedback and peer-feedback—MeMAD effectively pre-

Method	GPQA		Economics	
	GPT-4o	DeepSeek-V3	GPT-4o	DeepSeek-V3
Single Agent	0.490	0.523	0.798	0.768
MAD	0.540 (+0.050)	0.535 (+0.012)	0.796 (-0.002)	0.787 (+0.019)
MeMAD	**0.556 (+0.066)**	**0.540 (+0.017)**	**0.815 (+0.017)**	**0.815 (+0.047)**

Table 6: Generality evaluation of MeMAD across stronger models on GPQA and Economics datasets. Green text shows improvements over Single Agent, while gray indicates declines.

serves the rich iterative reasoning process that traditional MAD frameworks tend to discard. Our experiments on challenging tasks such as mathematical reasoning, scientific question answering, and language understanding demonstrate that MeMAD yields significant improvements in accuracy and robustness. In the future, we will investigate dynamic memory update strategies and explore the incorporation of external knowledge sources to further boost the system's reasoning capabilities.

Limitations

While MeMAD effectively reuses historical debate experiences to improve multi-step reasoning, it introduces additional overhead for maintaining, retrieving, and updating the structured memory. In particular, ensuring data quality in both positive and negative examples—along with their associated feedback—can be labor-intensive. Moreover, our approach assumes that tasks share enough similarity to benefit from prior debates, which may limit its impact on highly diverse or out-of-distribution tasks. Future work could explore more automated feedback generation, dynamic memory pruning, and adaptive retrieval techniques to further enhance MeMAD's scalability and generalization.

Acknowledgement

This research is supported by the National Research Foundation, Singapore under its National Large Language Models Funding Initiative (AISG Award No: AISG-NMLP-2024-002). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. This work is also supported by the National Natural Science Foundation of China under Joint Fund Project (U24A20328) and General Program (62476071), as well as the Guangdong Basic and Applied Basic Research Foundation (2025A1515011732).

References

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of thoughts: solving elaborate problems with large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 17682–17690, March 2024. doi: 10.1609/aaai.v38i16.29720.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, July 2020.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. ChatEval: Towards Better LLM-based Evaluators through Multi-Agent Debate, August 2023.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2318–2335. Association for Computational Linguistics, August 2024a. doi: 10.18653/v1/2024.findings-acl.137.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. ReConcile: Round-Table Conference Improves Reasoning via Consensus among Diverse LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 7066–7085. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.acl-long.381.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A Survey on In-context Learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1107–1128. Association for Computational Linguistics, November 2024. doi: 10.18653/v1/2024.emnlp-main.64.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving Factuality and Reasoning in Language Models through Multiagent Debate, 2023.

Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework, November 2024.

Sean Lee, Aamir Shakir, Darius Koenig, and Julius Lipp. Open source strikes bread - new fluffy embeddings model, 2024. URL https://www.mixedbread.ai/blog/mxbai-embed-large-v1.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, pp. 51991–52008, December 2023.

Xianming Li and Jing Li. AoE: Angle-optimized Embeddings for Semantic Textual Similarity. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1825–1839. Association for Computational Linguistics, August 2024. doi: 10.18653/v1/2024.acl-long.101.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. Improving Multi-Agent Debate with Sparse Communication Topology. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7281–7294. Association for Computational Linguistics, November 2024. doi: 10.18653/v1/2024.findings-emnlp.427.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate, 2023.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step, May 2023.

Tongxuan Liu, Xingyu Wang, Weizhe Huang, Wenjiang Xu, Yuting Zeng, Lei Jiang, Hailong Yang, and Jing Li. GroupDebate: Enhancing the Efficiency of Multi-Agent Debate Using Group Discussion, September 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. SELF-REFINE: Iterative refinement with self-feedback. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, pp. 46534–46594. Curran Associates Inc., May 2024.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. MetaICL: Learning to Learn In Context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2791–2809. Association for Computational Linguistics, July 2022. doi: 10.18653/v1/2022.naacl-main.201.

Ali Modarressi, Abdullatif Köksal, Ayyoob Imani, Mohsen Fayyaz, and Hinrich Schütze. MemLLM: Finetuning LLMs to Use An Explicit Read-Write Memory, January 2025.

OpenAI. GPT-4 Technical Report, March 2024.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as Operating Systems, February 2024.

Hongjin Qian, Peitian Zhang, Zheng Liu, Kelong Mao, and Zhicheng Dou. MemoRAG: Moving towards Next-Gen RAG Via Memory-Inspired Knowledge Discovery, September 2024.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A Graduate-Level Google-Proof Q&A Benchmark, November 2023.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language Agents with Verbal Reinforcement Learning, October 2023.

Qiushi Sun, Zhangyue Yin, Xiang Li, Zhiyong Wu, Xipeng Qiu, and Lingpeng Kong. Corex: Pushing the Boundaries of Complex Reasoning through Multi-Model Collaboration, August 2024.

Bo Wang, Heyan Huang, Yixin Cao, Jiahao Ying, Wei Tang, and Chong Feng. QRMeM: Unleash the Length Limitation through Question then Reflection Memory Mechanism. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4837–4851. Association for Computational Linguistics, November 2024a. doi: 10.18653/v1/2024.findings-emnlp.278.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-Agents Enhances Large Language Model Capabilities, June 2024b.

Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. Rethinking the Bounds of LLM Reasoning: Are Multi-Agent Discussions the Key? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6106–6131. Association for Computational Linguistics, August 2024c. doi: 10.18653/v1/2024.acl-long.331.

Xiaoqiang Wang, Suyuchen Wang, Yun Zhu, and Bang Liu. R$ˆ3$Mem: Bridging Memory Retention and Retrieval via Reversible Compression, February 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models, March 2023.

Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng Li, Xian Li, Bing Yin, Jingbo Shang, and Julian McAuley. MEMORYLLM: Towards self-updatable large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pp. 50453–50466. JMLR.org, July 2024d.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark, November 2024e.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent Abilities of Large Language Models, October 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, January 2023.

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

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A Survey of Large Language Models, October 2024.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731. AAAI Press, February 2024. doi: 10.1609/aaai.v38i17.29946.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models, April 2023.

A Experimental Details

In this section, we provide a detailed explanation of the implementation of MeMAD. The model is configured with a temperature parameter of 0.7, while other hyperparameters are set to their default values. Specifically, we employed the following model configurations: GPT-3.5-turbo-0125 (referred to as ChatGPT), GPT-4o-mini-2024-07-18 (referred to as GPT-4o-mini), and GPT-4o-2024-08-06 (referred to as GPT-4o) via the OpenAI API [1], Deepseek-V3 via the Deepseek API [2], as well as Qwen-2.5-14B-instruct-fp16 (referred to as Qwen2.5 14B) via the Alibaba Cloud API [3]. All MeMAD experiments were conducted with a configuration of three agents, with the maximum number of debate rounds set to 3. During the experience accumulation phase, all agents underwent the maximum number of debate rounds. In the retrieval and inference phase, the process terminated when all agents reached a consensus or the maximum debate rounds were reached. For the configurations of MAD and MoA experiments, we strictly adhered to the settings described in the original papers.

B Datasets

We conducted experiments on datasets covering a wide range of domains, including complex reasoning problems in mathematics, physics, chemistry, biology, law, and economics.

[1] https://openai.com/
[2] https://www.deepseek.com/
[3] https://www.aliyun.com/

- **MATH500**: This dataset is a subset of the MATH benchmark, consisting of 500 mathematical problem-solving questions categorized by topic and difficulty (Lightman et al., 2023). In our study, the training set was constructed by randomly sampling 100 problems per category from the MATH training set, restricted to problems with a difficulty level of at least 3. For evaluation, to ensure the challenge of the experiments, we selected 134 problems with the highest difficulty level (level 5) from the MATH500 dataset as the evaluation set.
- **GPQA**: A dataset comprising 448 graduate-level multiple-choice questions spanning biology, physics, and chemistry. All questions were validated by domain experts to ensure both objectivity and difficulty (Rein et al., 2023). For our experiments, we used the GPQA Diamond subset (198 questions) as the test set, while the remaining questions from the GPQA Main set (excluding GPQA Diamond) were used as the training set.
- **MMLUPro-Law**: MMLUPro is a comprehensive benchmark designed for multi-discipline language understanding and reasoning, spanning 14 domains (Wang et al., 2024e). For this study, we utilized the Law subset, which contains 1,101 high-quality multiple-choice questions. The dataset was randomly divided into training and testing sets in a 3:1 ratio, resulting in 826 questions for experience accumulation and 276 for evaluation.
- **MMLUPro-Economics**: Similarly, we also employed the Economics subset of MMLUPro (Wang et al., 2024e), which comprises 844 multiple-choice questions. Following the same 3:1 split strategy, we obtained 633 questions for training and 211 for testing.

C Baseline Details

Here we list more details about baseline models:

- Self-Refine. It generates an initial output with an LLM, then the LLM gives feedback and refines the output iteratively. It doesn't need extra training data or RL, and performs well across diverse tasks, serving as a benchmark for comparison (Madaan et al., 2024).
- Self-Consistency: It samples diverse reasoning paths from a language model's decoder, then aggregates final answers by marginalizing out the reasoning paths. This unsupervised method improves language models' reasoning performance on various tasks (Wang et al., 2023).
- Reflexion: It reinforces language agents via linguistic feedback. Agents reflect on task feedback, store reflective text in memory, and use it for better decision-making. It shows good performance in diverse tasks like decision-making, coding, and language reasoning (Shinn et al., 2023).
- MAD: This approach generates answers through multi-round debates among multiple language model instances, which can improve reasoning and performs well in various tasks, providing a reference for comparative experiments (Du et al., 2023).
- MoA: It employs a layered architecture, with each layer comprising multiple LLM agents. Agents use outputs from the previous layer as supplementary input, enabling effective information flow. MoA has demonstrated notable performance across a diverse range of tasks (Wang et al., 2024b).

D Additional Experiments

D.1 The impact of memory on different models

Memory boosts reasoning for single and multi-agent models. To systematically evaluate the effectiveness of memory augmentation, we conducted comparative experiments on both single-agent and multi-agent baselines, with and without memory augmentation. For a fair

comparison, we implemented a memory-augmented single-agent framework that mirrors our approach in memory retrieval mechanism—incorporating both successful and failed experiences from the training set through semantic similarity matching.

Figure 3 presents the comparative results across different model configurations. The experimental outcomes demonstrate that memory augmentation consistently enhances reasoning performance across all agent architectures while maintaining parameter-free operation. Notably, more advanced models like GPT-4o-mini and Qwen2.5 exhibit more substantial improvements compared to ChatGPT, suggesting that stronger models possess superior capabilities in leveraging structured experiences. Furthermore, the performance gap between MeMAD and MAD significantly exceeds that between Me+MoA and MoA, indicating that while these accumulated experiences can enhance MoA's performance, their effectiveness is particularly pronounced in debate-based frameworks where interaction plays a crucial role.

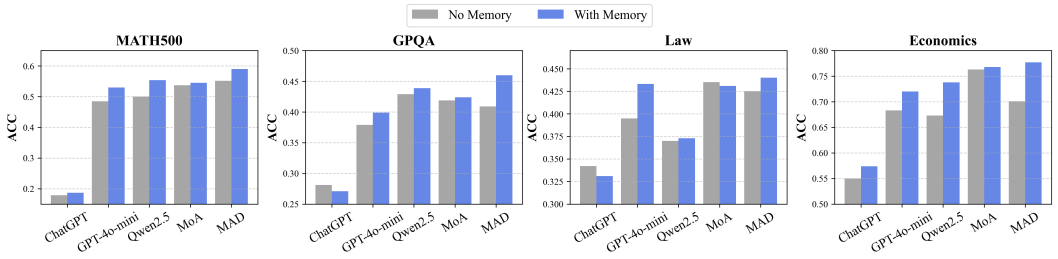

Figure 3: Performance of different methods with and without memory augmentation.

D.2 Token Usage Analysis

MeMAD enhances reasoning ability without increasing computational overhead. This experiment investigates the average token consumption per question for MAD and MeMAD across various datasets, analyzing both prompt and completion token usage. As illustrated in Table 7, MeMAD integrates structured memory into prompts, yet its overall token consumption remains comparable to or even lower than that of MAD. Notably, MeMAD consistently reduces completion token usage. This efficiency is achieved because MeMAD terminates the debate process as soon as all agents reach a consensus, avoiding unnecessary token generation in prolonged interactions.

Method	#Prompt Tokens		#Completion Tokens	
	MAD	MeMAD	MAD	MeMAD
GPQA	17.81K	16.56K	3.61K	2.49K
MATH500	20.96K	25.39K	5.76K	4.11K
Law	12.48K	19.66K	1.65K	1.64K
Economics	10.06K	7.67K	1.68K	0.87K

Table 7: Comparison of token consumption between MAD and MeMAD.

E Prompts in MeMAD

E.1 Prompts in Dual-level Feedback Mechanism

The prompts for the self-feedback module and the peer-feedback module are shown in Table 8 and Table 9, respectively.

E.2 Prompts for different Tasks

The prompts for different tasks are shown in Table 10.

Self-Feedback Prompt Template

You are tasked with analyzing a problem-solving example and generating transferable insights for future improvement on solving similar problems.

Given Information:

<question> {{question}} </question>

<correct_solution> {{correct_solution}} </correct_solution>

<llm_response> {{llm_response}} </llm_response>

<response_correctness>

Response Correctness is correct [or incorrect].

</response_correctness>

Analysis Framework:

1. Compare the solutions: 1) Identify similarities and differences in approach; 2) Analyze which elements worked better and why; 3) Note any efficiency or clarity advantages

2. If correct: 1) Identify successful reasoning patterns; 2) Extract key decision points that led to success

3. If incorrect: 1) Identify the gap between response and correct answer; 2) Analyze where the reasoning went wrong. 3) Determine what knowledge or step was missing

Output Requirements:

Generate exactly 3 key learned insights in this format:

"Key Learning #[number]: [specific insight]

Application: [how to apply this learning to future problems]"

Each learned insight must be: 1) Generalizable (applicable to similar problems) 2) Specific (clear action or thinking strategy) 3) Concise (one sentence for insight, one for application) 4) Focus on problem-solving strategies only

Note:

1) Do not restate the specific problem content or solution; 2) Only output the learned insights.

Table 8: Prompt template for self-feedback.

Peer-Feedback Prompt Template

You are tasked with analyzing an incorrect LLM response by comparing it with both the standard solution and a correct LLM response, to generate generalizable insights that can help LLMs better solve similar problems in the future.

Given Information:

<question> {{question}} </question>

<correct_solution> {{correct_solution}} </correct_solution>

<correct_llm_response> {{correct_llm_response}} </correct_llm_response>

<incorrect_llm_response> {{incorrect_llm_response}} </incorrect_llm_response>

Analysis Framework:

1. Error Pattern Analysis: 1) Identify where incorrect response deviates from both correct approaches; 2) Analyze the root causes of these deviations; 3) Detect patterns of misconceptions or flawed reasoning.

2. Success Pattern Recognition: 1) Study how correct LLM response aligns with standard solution; 2) Identify key elements missing in incorrect response; 3) Extract successful reasoning patterns and approaches.

3. Improvement Opportunities: 1) Pinpoint specific areas where incorrect response could be enhanced; 2) Identify critical checkpoints that could prevent similar errors; 3) Formulate strategies to bridge the gap between incorrect and correct approaches.

Output Requirements:

Generate exactly 3 transferable insights in this format: "Learning Point [number]: [specific insight]; Strategic Application: [concrete strategy for future problem-solving]"

Each insight must be: 1) Focused on preventing similar errors in future; 2) Strategy-focused (not problem-specific); 3) Action-oriented; 4) Clearly articulated in 1-2 sentences.

Note:

1) Emphasize practical strategies for enhancement; 2) Ensure insights are applicable to future problem-solving; 3) Avoid repeating specific problem details or solutions; 4) Emphasize methodological improvements rather than content knowledge; 5) Only output the learned insights.

Table 9: Prompt template for peer-feedback.

Prompts for different Tasks
MATH500:
Can you solve the following math question as accurately as possible?
<question>{{question}}</question>
Present your analysis concisely using only essential reasoning steps. Provide the final answer in double parentheses at the end of your response : ((answer))
GPQA:
Please analyze the following question and solve it as accurately as possible.
<question>{{question}}</question>
Present your analysis concisely using only essential reasoning steps. Provide the final answer in double parentheses at the end of your response: ((answer)), where answer is A, B, C, or D.
MMLUPro-Law:
Please analyze the following legal question:
<question>{{question}}</question>
Present your analysis concisely using only essential reasoning steps and select the correct answer. Provide your final answer in double parentheses: ((answer)), where answer can be A, B, C, D, E, F, G, H, I, or J.
MMLUPro-Economics:
Please analyze the following economics question:
<question>{{question}}</question>
Present your analysis concisely using only essential reasoning steps and select the correct answer. Provide your final answer in double parentheses: ((answer)), where answer can be A, B, C, D, E, F, G, H, I, or J.

Table 10: Prompt template for different task.