MULTI-AGENT DEBATE WITH MEMORY MASKING

Anonymous authors
Paper under double-blind review

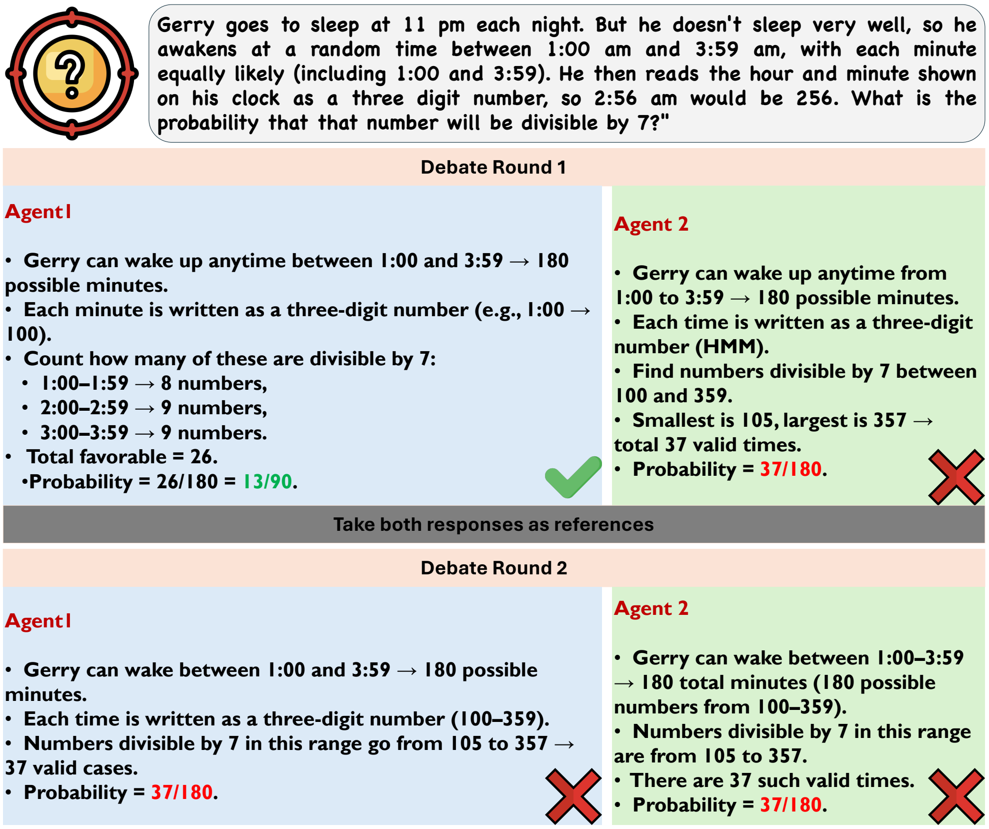
ABSTRACT

Large language models (LLMs) have demonstrated impressive capabilities in various language-based reasoning tasks (e.g., math reasoning). Among all LLM reasoning frameworks, *multi-agent debate* (MAD), which employs multiple LLM agents and performs reasoning in the way of multi-round debate, has emerged as a powerful reasoning paradigm since it allows agents to access previous memories to refine their reasoning iteratively in each debate round and facilitates LLMs in alleviating the potential intrinsic self-preference bias. Although MAD improves the reasoning capabilities of LLMs significantly, in this paper, however, we theoretically demonstrate that the performance of MAD is closely related to the quality of memories. This indicates that MAD remains vulnerable to erroneous reasoning memories, which poses a threat to the robustness of MAD. To address this problem, we introduce a simple yet effective multi-agent debate framework, *multi-agent debate with memory masking* (MAD-M^2), to enhance the robustness of MAD by allowing LLM agents to select memories in the previous debate round before they perform reasoning in the current debate round. In this way, MAD-M^2 can polish the contextual information at the beginning of each debate round by preserving as many informative and meaningful memories as possible while dropping the erroneous memories and, in turn, achieve better reasoning performance. Extensive evaluations on mainstream mathematical and logical reasoning benchmarks demonstrate that MAD-M^2 can achieve better reasoning performance than MAD.

1 INTRODUCTION

Large language models (LLMs) have shown impressive instruction-following capabilities in language understanding and question answering. In particular, reasoning is a typical case among various capabilities and has attracted more and more attention. Specifically, in LLMs, given detailed cognitive instruction (Ouyang et al., 2022), comprehensive reasoning behaviors can be effectively elicited. Moreover, it has been demonstrated that the reasoning capability can be further enhanced (e.g., step-by-step reasoning (Wei et al., 2022; Kojima et al., 2022), task decomposition (Zhou et al., 2023), and reflection (Madaan et al., 2023)) by feeding LLMs with some demonstrations (e.g., In-Context Learning). Notably, the test-time scaling law (Snell et al., 2024; Brown et al., 2024) of LLMs exhibits the potential to further enhance the reasoning with repeated sampling (Wang et al., 2023; Taubenfeld et al., 2025), which provides a simple yet powerful reasoning strategy in complex problems. Despite such an impressive capability, the reasoning performed via individual models is still faced with notable limitations, especially the insufficient generation diversity problem (Si et al., 2025; Hayati et al., 2023), where multiple samplings are limited to a single reasoning method, which reduces the effectiveness of scaling up sampling and restricts the practical application of the scaling up concept.

Recently, inspired by the theory of *The Society of Mind* (Minsky, 1986), the *multi-agent debate* (MAD (Du et al., 2023)) framework has emerged as a promising approach to overcome the limitations of the single LLM reasoning paradigm. In this framework, multiple LLMs are employed as agents to perform reasoning on a given question in a multi-round debate. Specifically, in each debate round (except for the first round), LLM agents are allowed to observe their memories in the previous debate round and are required to propose candidate answers based on their critical evaluations of all previous memories. Compared to the single LLM reasoning paradigm, the multi-agent debate framework improves the reasoning capabilities mainly from two perspectives. On the one hand, in such a multiple-agent and multiple-round reasoning framework, LLMs have access to all memories of the previous round and are required to generate new answers based on their critical thinking on these



Gerry goes to sleep at 11 pm each night. But he doesn't sleep very well, so he awakens at a random time between 1:00 am and 3:59 am, with each minute equally likely (including 1:00 and 3:59). He then reads the hour and minute shown on his clock as a three digit number, so 2:56 am would be 256. What is the probability that that number will be divisible by 7?"

- Gerry can wake up anytime between I:00 and 3:59 → I80 possible minutes.
- Each minute is written as a three-digit number (e.g., I:00 → I00).
- Count how many of these are divisible by 7:
  - I:00–I:59 → 8 numbers,
  - 2:00–2:59 → 9 numbers,
  - 3:00–3:59 → 9 numbers.
- Total favorable = 26.
- Probability = 26/I80 = I3/90.

- Gerry can wake up anytime from I:00 to 3:59 → I80 possible minutes.
- Each time is written as a three-digit number (HMM).
- Find numbers divisible by 7 between I00 and 359.
- Smallest is I05, largest is 357 → total 37 valid times.
- Probability = 37/I80.

- Gerry can wake between I:00 and 3:59 → I80 possible minutes.
- Each time is written as a three-digit number (I00–359).
- Numbers divisible by 7 in this range go from I05 to 357 → 37 valid cases.
- Probability = 37/I80.

- Gerry can wake between I:00–3:59 → I80 total minutes (I80 possible numbers from I00–359).
- Numbers divisible by 7 in this range are from I05 to 357.
- There are 37 such valid times.
- Probability = 37/I80.

Figure 1: An illustration of the effect of wrong memories. The example is derived from DeepSeek-V3 with a case in the MATH dataset. In the conventional multi-agent debate framework, all memories in the previous debate round are taken into consideration in the next debate round. However, the memories from the previous round may include wrong reasoning responses (see the responses of debate round 1 in the figure). If the incorrect memory is referred in the next round, the agent, which originally correctly performs the reasoning may be misled (see Agent 1 in the debate round 2).

memories. Thus, the debate process can be treated as a refinement of reasoning, which facilitates achieving better reasoning results. On the other hand, it has been demonstrated that there exists a *self-preference* phenomenon (Panickssery et al., 2024) that LLMs tend to assign higher scores to answers generated by themselves, though all answers are considered as equal quality by human annotators. Such a preference may potentially amplify the bias of LLMs to their own memories and subsequently result in incorrect reasoning (Xu et al., 2024). Thus, by applying the multi-agent debate framework, the phenomenon can be alleviated through the critical evaluations of memories of LLMs.

However, a concern about the typical MAD framework is that LLM agents are not robust to the erroneous memories. Specifically, as shown in Fig. 1, the memories derived in the previous debate round probably contain incorrect reasoning paths (e.g., Agent 2 in the debate round 1), and the agent, which takes all memories into consideration, will be misled and generate the incorrect answers (e.g., Agent 1 in debate round 2). Such a phenomenon potentially undermines the performance of MAD.

For a better understanding, in this paper, we first theoretically demonstrate that the conventional multi-agent debate framework is vulnerable to erroneous memories. Specifically, the performance of LLM agents in a debate round is closely related to the number of wrong memories in the previous debate round. When the number of wrong memories increases, the capability decreases correspondingly. According to our theoretical analyses, in both easy and hard reasoning scenarios, enhancing the robustness of LLM agents to erroneous memories consistently improves the performance of MAD.

Motivated by these theoretical and empirical observations, we propose a simple yet effective method, *multi-agent debate with memory masking* (MAD-M^2), to enhance the robustness of the MAD framework to erroneous memories by allowing LLMs to mask the wrong memories in the previous debate round based on their evaluations. Specifically, at each debate round (except for the first debate round), LLM agents are required to critically evaluate and score the memories in the previous debate round. Then, a binary mask vector will be generated according to the scores to filter the memories. The number of preserved memories varies across agents and tasks. Then, LLM agents are required to generate new responses based on the preserved memories. The final answer will be selected from a set of responses by performing majority voting in the final round. Intuitively, the initial round of the multi-agent debate can be viewed as zero-shot reasoning, while the latter rounds are equivalent to in-context learning. Specifically, in the initial round, responses are generated merely with the query. However, in the latter rounds, responses are generated with both the query and the memories from the previous round. From this perspective, the reason that performing memory masking helps improve the performance of MAD can, to some extent, be attributed to context engineering (Mei et al., 2025), which aims at polishing the quality of contextual information in prompts by incorporating critical information while filtering out trivial content based on the specific question-answering objectives.

To validate the effectiveness of our proposed MAD-M^2, four mainstream mathematical reasoning and language understanding benchmarks, which are MATH (Hendrycks et al., 2021), MMLU-Pro (Wang et al., 2024), AIME24, and AIME25, are adopted for evaluation. According to the empirical results, MAD-M^2 can achieve better performance than the conventional MAD framework (Du et al., 2023) on easy reasoning tasks with weak LLMs. Moreover, the empirical results also demonstrate our analysis that enhancing the capability/robustness of agents benefits the performance of MAD in Section 2.2.

Overall, our contribution in this paper can be primarily summarized as the following several aspects:

- We find that LLM agents in the conventional multi-agent debate framework are vulnerable to wrong memories and may be misled to generate incorrect reasoning responses (c.f. Fig. 1). Further, we demonstrate the vulnerability of the MAD framework from the theoretical perspective in Section 2.2, and find that performing memory masking is a feasible way to enhance the performance of the MAD framework. As far as we know, we are the first work to discuss the phenomenon in the MAD.
- To alleviate the detriment derived from the wrong memories, we propose a simple yet effective multi-agent debate framework, multi-agent debate with memory masking (MAD-M^2), to enhance the capability of the multi-agent debate framework by allowing LLM agents to critically evaluate memories and mask the undesirable cases for further reasoning in the next debate round in Section 3.
- Our empirical evaluations on four mainstream reasoning benchmarks demonstrate the effectiveness of our proposed MAD-M^2 can achieve better performance than MAD in most cases. This further indicates that MAD-M^2 facilitates the capability of the reasoning framework (c.f. Section 4).

2 An Overview of Multi-Agent Debate Framework

In this section, we first introduce an overview of the typical multi-agent debate framework (Du et al., 2023) and formulate the workflow of the MAD framework. Then, we theoretically analyze the probability that CoT-SC (Wang et al., 2022) and MAD successfully infer the answer to the query and further show that the naive MAD framework is vulnerable to wrong memories in the previous debate.

2.1 Problem Formulation of Multi-Agent Debate

Multi-agent debate (MAD) (Du et al., 2023) is a powerful reasoning paradigm that infers the answers to questions through multiple rounds of interactions among multiple LLM agents. Specifically, in the typical multi-agent debate framework, LLMs are first instructed to perform reasoning on the queries independently at the initial round. Then, from the second round, agents are required to critically evaluate the memories of all agents in the last debate round to generate new responses. After a multi-round debate, the final answer is obtained when a consensus is reached among all LLM agents.

Consider a typical multi-agent debate framework (Du et al., 2023) composed of a total of N_{round} debate rounds and a set of N_{a} LLM agents $\mathcal{A} = \{A_{\theta_1}, A_{\theta_2}, ..., A_{\theta_{N_{\text{a}}}}\}$ that are parameterized with parameters θ_i, respectively. Consider a query prompt $x^{\text{test}} \in \mathcal{X}$ that includes both task instructions and the question, where $\mathcal{X} = \{x_1^{\text{test}}, x_2^{\text{test}}, ..., x_{\text{N}_{\text{t}}}^{\text{test}}\}$ is a discrete set of query prompts. Let $\mathcal{M}_r = [A_{\theta_1}(x^{\text{test}}, \mathcal{M}_{r-1}), A_{\theta_2}(x^{\text{test}}, \mathcal{M}_{r-1}), ..., A_{\theta_{N_{\text{a}}}}(x^{\text{test}}, \mathcal{M}_{r-1})]$, where $1 < r < N_{\text{round}}, \mathcal{M}_1 = \emptyset$, denote the memory set derived from the r-th debate round. Then, for each LLM agent $A_{\theta_i} \in \mathcal{A}$, given the memories $\mathcal{M}_r$ and prompt x^{test}, the response generation in the next round is formulated as:

$$A_{\theta_i}(x^{\text{test}}, \mathcal{M}_r) = \hat{x}_{1:L}, \text{where } \hat{x}_l = \arg\max_x P(x|\hat{x}_{<l}; x^{\text{test}}; \mathcal{M}_r; \theta_i), \tag{1}$$

where $\hat{x}_l$ denotes the l-th word prediction of the response, $\hat{x}_{<l}$ denotes the previous word predictions of $\hat{x}_l$, and L denotes the length of the response. The response $\hat{x}_{1:L}$ is a sequence with the length of L.

2.2 Vulnerability of MAD to Wrong Memories

In the typical multi-agent debate framework, as shown in Eq. (1), LLM agents are required to generate new reasoning responses based on the memories derived from the previous debate round. Thus, intuitively, erroneous memories in the previous debate round may potentially damage the performance of MAD. However, as far as we know, few works have discussed the issue in detail. To figure out this issue, we provide analyses in the following to explore the negative effect of erroneous memories.

2.2.1 A Theoretical Perspective for Understanding of MAD

As a comparison, we select CoT-SC (Wang et al., 2022) as the counterpart of MAD. From the perspective of the reasoning paradigm, both CoT-SC and MAD can be seen as test-time scaling reasoning (Snell et al., 2024; Brown et al., 2024), which improves the robustness and the performance of reasoning by applying multiple response generations for a single query. The main difference between the two reasoning paradigms is that CoT-SC performs reasoning once and obtains the answer by voting for the majority among a set of responses, while MAD is formulated as multi-round reasoning, where the final answer is voted from a set of responses generated in the final debate round.

Assumption 2.1. *Assume that the probability that an agent independently generates the correct answer only with the given query is $p \in [0, 1]$, while the probability that the agent generates correct answers to the query based on the previous memories is assumed to be $e^{-\alpha N_{\mathrm{e}}}$, where N_{e} denotes the number of erroneous memories and $\alpha \in \mathbb{R}^+$ is a consistent coefficient that indicates the robustness of agents to erroneous memories. The reasoning is deemed as successful when the number of correct answers N_{cor} in the final round satisfies $N_{\mathrm{cor}} > \frac{N_{\mathrm{res}}}{2}$, where N_{res} denotes the number of responses.*

Remark 1. *In Assumption 2.1, we formulate the probability that a single LLM agent generates the correct answer to the given query based on the previous memories as $e^{-\alpha N_{\mathrm{e}}}$, where $N_{\mathrm{e}} \in \{0, 1, 2, ..., N_{\mathrm{a}}\}$ denotes the number of erroneous memories. Two aspects are considered here. On the one hand, in an ideal case where the number of erroneous memories is 0, the probability of the agent generating the correct reasoning in the next debate round will be 1. On the other hand, in the case where all memories are erroneous (i.e., $N_{\mathrm{e}} = N_{\mathrm{a}}$), the probability of the agent generating a correct response to the query will degrade to $e^{-\alpha N_{\mathrm{a}}} > 0$, which aligns with the intuition that agents can retain a non-zero probability of generating correct reasoning even if all memories are erroneous.*

Proposition 2.2 (**CoT-SC**). *Consider a total number of N_{sc} independent responses generated in the way of CoT-SC. With Assumption 2.1, the probability that the final answer is correct is bounded by:*

$$
\begin{aligned}
P(N_{\mathrm{cor}} > \frac{N_{\mathrm{sc}}}{2}) &\leq \exp\left(-2N_{\mathrm{sc}}(\frac{1}{2} - p)^2\right), \qquad p < \frac{1}{2}, \\
P(N_{\mathrm{cor}} > \frac{N_{\mathrm{sc}}}{2}) &\geq 1 - \exp\left(-2N_{\mathrm{sc}}(\frac{1}{2} - p)^2\right), \quad p \geq \frac{1}{2}.
\end{aligned}
$$

The corresponding lower bound and upper bound of cases $p < \frac{1}{2}$ and $p \geq \frac{1}{2}$ are 0 and 1, respectively.

Proposition 2.3 (**MAD**). *Consider a 2-round MAD reasoning, where N_{a} agents are involved in each debate round. With Assumption 2.1, the probability that the final answer is correct is bounded by:*

$$
\begin{aligned}
P(N_{\mathrm{cor}} > \frac{N_{\mathrm{a}}}{2}) &\leq \sum_{j=0}^{N_{\mathrm{a}}} \binom{N_{\mathrm{a}}}{j} p^j (1-p)^{N_{\mathrm{a}}-j} \exp\left(-2N_{\mathrm{a}}\left(\frac{1}{2} - e^{\alpha(j-N_{\mathrm{a}})}\right)^2\right), \qquad e^{\alpha(j-N_{\mathrm{a}})} < \frac{1}{2}, \\
P(N_{\mathrm{cor}} > \frac{N_{\mathrm{a}}}{2}) &\geq \sum_{j=0}^{N_{\mathrm{a}}} \binom{N_{\mathrm{a}}}{j} p^j (1-p)^{N_{\mathrm{a}}-j} \left(1 - \exp\left(-2N_{\mathrm{a}}\left(\frac{1}{2} - e^{\alpha(j-N_{\mathrm{a}})}\right)^2\right)\right), \quad e^{\alpha(j-N_{\mathrm{a}})} \geq \frac{1}{2}.
\end{aligned}
$$

The corresponding lower and upper bounds of cases $e^{\alpha(j-N_{\mathrm{a}})} < \frac{1}{2}$ and $e^{\alpha(j-N_{\mathrm{a}})} \geq \frac{1}{2}$ are 0 and 1.

Remark 2. *The proofs of Propositions 2.2 and 2.3 are available in Appendix B. In Propositions 2.2 and 2.3, we investigate the bounds of the probability of CoT-SC and MAD correctly generating correct reasoning responses to a query based on the workflow of their pipelines. In the case of CoT-SC, we can observe that the performance is mainly determined by the number of generated responses (i.e., N_{sc}) and the capability of the LLM agents in reasoning (i.e., the probability of LLM agents correctly answering the query p). However, in the case of MAD, although the performance is explicitly determined by the number of agents (i.e., N_{a}) and the probability of LLM agents generating correct answers based on the memories in the previous round (i.e., $e^{-\alpha(N_{\mathrm{a}}-j)}$), some other aspects also implicitly influence the reasoning performance of the MAD framework due to the multi-round* *reasoning paradigm. On the one hand, the probability $e^{-\alpha(N_{\mathrm{a}}-j)}$ is determined by the number of erroneous memories in the previous round. On the other hand, the performance is also influenced by the probability distribution of the memories in the previous debate round. In our 2-round MAD analysis, the distribution is formulated as a binomial distribution: $P(X=j)=\binom{N_{\mathrm{a}}}{j}p^j(1-p)^{N_{\mathrm{a}}-j}$.*

According to Propositions 2.2 and 2.3, we can observe that the bounds of the probability that MAD correctly generates correct reasoning share a similar format with those of CoT-SC. In both cases, the bounds are closely related to the number of responses and the capability of a single LLM agent in handling the given reasoning task. However, the capability of agents in the MAD framework is determined by the quality of memories generated in the previous debate round. Consider a case, where $N_{\mathrm{a}} = N_{\mathrm{sc}}$. With the assumption that $e^{\alpha(j-N_{\mathrm{a}})}$ does not deviate too much from p (e.g., α is small), since the probability $\binom{N_{\mathrm{a}}}{j}p^j(1-p)^{N_{\mathrm{a}}-j} \leq 1$ in all cases, the performance of MAD can hardly be superior to CoT-SC. This observation, to some extent, contributes to explaining why MAD empirically fails to outperform CoT-SC in reasoning (Huang et al., 2023). Moreover, in the case where more erroneous memories are involved (i.e., $j \to 0$), the probability $e^{\alpha(j-N_{\mathrm{a}})}$ drops significantly. Correspondingly, both the upper and lower bounds will then shrink exponentially. This phenomenon indicates that erroneous memories tend to deteriorate the performance of MAD. Thus, we find that MAD is vulnerable to erroneous memories. More discussions are provided in the following section.

2.2.2 Further Discussions about the Theoretical Results

In both CoT-SC and MAD, the bounds of the probability that the answer to the given query is correctly inferred are discussed in two cases. Specifically, when $p < \frac{1}{2}$ or $e^{-\alpha N_{\mathrm{e}}} < \frac{1}{2}$, the probability is upper bounded, while the probability is lower bounded when $p \geq \frac{1}{2}$ or $e^{-\alpha N_{\mathrm{e}}} \geq \frac{1}{2}$. Both cases implicitly correspond to two types of reasoning cases: *hard problem reasoning* and *easy problem reasoning*.

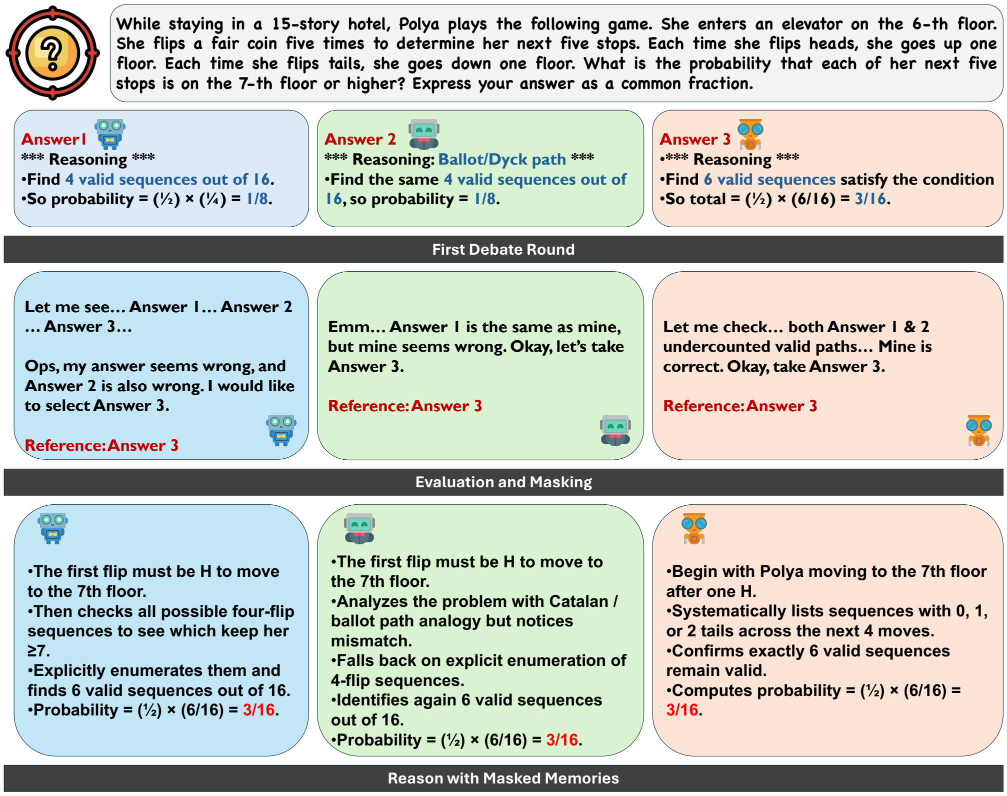
Hard Problem Reasoning. In the context of hard problem reasoning (HPR) settings, LLM agents can hardly correctly infer the answer to the query due to the difficulty of the problem and the erroneous memories in the previous reasoning round. Such a case corresponds to the case of $p < \frac{1}{2}$ or $e^{-\alpha N_{\mathrm{e}}} < \frac{1}{2}$. In CoT-SC, when the capability of LLM agents (i.e., the probability p) is fixed, increasing the number of responses does not contribute to improving performance. In contrast, when the number of responses (i.e., N_{sc}) is fixed, enhancing the capability of LLM agents ($p \to \frac{1}{2}$) will improve the reasoning performance. Meanwhile, in MAD, we can obtain similar observations. In the 2-round multi-agent debate case, the answers are independently generated from multiple LLM agents at the final round of debate. However, memories from the previous round will affect the probability of LLM agents correctly answering the query in the final round. By treating the probability of all reasoning cases in the first round as a constant, we can observe that increasing the number of agents (i.e., N_{a}) does not contribute to improving the reasoning performance while enhancing the robustness of agents to wrong memories (i.e., $e^{\alpha(j-N_{\mathrm{a}})} \to \frac{1}{2}$) facilitates improving the performance of MAD.

Easy Problem Reasoning. In contrast to hard problem reasoning settings, LLM agents can easily handle the query correctly in the context of easy problem reasoning settings (EPR). Such a case corresponds to the case of $p \geq \frac{1}{2}$ or $e^{-\alpha N_{\mathrm{e}}} \geq \frac{1}{2}$. In both CoT-SC and MAD frameworks, we can observe that either increasing the number of agents (i.e., N_{sc} or N_{a}) or enhancing the robustness of the LLM agents (i.e., $p \to 1$ or $e^{\alpha(j-N_{\mathrm{a}})} \to 1$) improves the performance of the reasoning paradigms.

Thus, with both hard and easy problem reasoning settings above taken into consideration, we find that a feasible way to improve the performance of the multi-agent debate framework is to improve the robustness of LLM agents. In the context of our analysis framework, the probability that LLM agents generate correct answers to the query based on the memories in the previous round is influenced by two aspects: the capability of the LLM agents p and the number of erroneous memories in the previous round $N_{\mathrm{a}} - j$. Since the capability of LLMs is determined by the pre-training, the robustness of agents can be improved by reducing the number of erroneous memories in the previous round.

3 MAD-M^2: Multi-Agent Debate with Memory Masking

In Section 2, we have demonstrated that a feasible way to improve the performance of the multi-agent debate framework is to improve the robustness of LLM agents by removing the wrong memories in the previous round. To this end, we propose a simple yet effective multi-agent framework, *multi-agent debate with memory masking* (MAD-M^2), to allow LLM agents to mask wrong memories in the previous debate round and perform reasoning with the remaining memories in the next debate round.



While staying in a 15-story hotel, Polya plays the following game. She enters an elevator on the 6-th floor. She flips a fair coin five times to determine her next five stops. Each time she flips heads, she goes up one floor. Each time she flips tails, she goes down one floor. What is the probability that each of her next five stops is on the 7-th floor or higher? Express your answer as a common fraction.

Answer1
*** Reasoning ***
•Find 4 valid sequences out of 16.
•So probability = (½) × (¼) = 1/8.

Answer 2
*** Reasoning: Ballot/Dyck path ***
•Find the same 4 valid sequences out of 16, so probability = 1/8.

Answer 3
•*** Reasoning ***
•Find 6 valid sequences satisfy the condition
•So total = (½) × (6/16) = 3/16.

Let me see… Answer 1… Answer 2 … Answer 3…

Ops, my answer seems wrong, and Answer 2 is also wrong. I would like to select Answer 3.

Reference: Answer 3

Emm… Answer 1 is the same as mine, but mine seems wrong. Okay, let's take Answer 3.

Reference: Answer 3

Let me check… both Answer 1 & 2 undercounted valid paths… Mine is correct. Okay, take Answer 3.

Reference: Answer 3

•The first flip must be H to move to the 7th floor.
•Then checks all possible four-flip sequences to see which keep her ≥7.
•Explicitly enumerates them and finds 6 valid sequences out of 16.
•Probability = (½) × (6/16) = 3/16.

•The first flip must be H to move to the 7th floor.
•Analyzes the problem with Catalan / ballot path analogy but notices mismatch.
•Falls back on explicit enumeration of 4-flip sequences.
•Identifies again 6 valid sequences out of 16.
•Probability = (½) × (6/16) = 3/16.

•Begin with Polya moving to the 7th floor after one H.
•Systematically lists sequences with 0, 1, or 2 tails across the next 4 moves.
•Confirms exactly 6 valid sequences remain valid.
•Computes probability = (½) × (6/16) = 3/16.

Figure 2: An illustration of MAD-M^2 framework. In general, MAD-M^2 mainly includes three steps. (i) In the initial debate round, LLM agents independently generate responses based on the given query. (ii) The responses generated in the previous round are treated as memories. Each LLM agent is required to evaluate all memories critically and mask the undesirable memories for the reasoning in the next debate round. (iii) With the masked memories, agents perform reasoning in the next round.

3.1 Detailed Implementation of MAD-M^2

An overview of our proposed MAD-M^2 framework is visualized in Fig. 2. In general, the core idea of the proposed MAD-M^2 framework is inserting an evaluation and masking operation between two reasoning rounds to reduce the potential wrong memories. In this way, it is equivalent to enhancing the capability of agents in generating correct answers based on the memories in the previous round.

Step 1. Initial Debate Round. In the MAD-M^2 framework, at the initial debate round, all N_{a} LLM agents are required to independently generate responses to the given query $x^{\text{text}} \in \mathcal{X}$ without any contextual information. Specifically, a total of N_{a} responses are independently generated from LLM agents $A_{\theta_i} \in \mathcal{A}$, respectively, to formulate the memory vector of the initial round: $\mathcal{M}_1 = [A_{\theta_1}(x^{\text{test}}, \emptyset), A'_{\theta_2}(x^{\text{test}}, \emptyset), ..., A_{\theta_{N_{\mathrm{a}}}}(x^{\text{test}}, \emptyset)]$, where $A'_{\theta_i}(x^{\text{test}}, \emptyset)$ denotes the wrong reasoning responses generated by the i-th LLM agent while $A_{\theta_i}(x^{\text{test}}, \emptyset)$ denotes correct reasoning responses.

Step 2. Evaluation and Masking. After a round of debate, a set of memories regarding the given query is obtained. Here, we consider a case where both correct and incorrect reasoning results are contained. Specifically, the memories can be completely correct, partially correct, or completely wrong. According to our previous demonstration, in the case where wrong memories are included, the capability of agents to generate correct responses in the next debate round will be undermined. To solve this problem, in MAD-M^2, we propose to allow each agent to critically evaluate the memories from the previous debate round and generate a binary vector $M = \{0, 1\}^{N_{\mathrm{a}}}$ to mask wrong memories identified by the agent. Then, we can obtain a new set of memories with the wrong memories reduced:

$$\widehat{\mathcal{M}}_r^{(i)} = M^{(i)} \odot \mathcal{M}_r, \text{where } M^{(i)} = s \circ A_{\theta_i}(\mathcal{M}_r), \tag{2}$$

where $s \circ A_{\theta_i}(\mathcal{M}_r) \mapsto \{0, 1\}^{N_{\mathrm{a}}}$ denotes an operation that maps the evaluation of agent A_{θ_i} over the memory $\mathcal{M}_r$ obtained from the r-th debate round into a binary mask vector, $\widehat{\mathcal{M}}_r^{(i)}$ denotes the new set of memories selected by the agent from the memory, and $\odot$ denotes the element-wise multiplication.

Subjective Masking Strategy. In the subjective masking strategy, memories are masked according to the subjective evaluations of LLM agents. Specifically, in MAD-M^2, LLM agents are required to evaluate memories with the flags “YES”, “NO”, and “NOT SURE”. Depending on the strictness of the predefined filtering rule, the “NOT SURE” is treated as either “YES” or “NO” correspondingly.

Objective Masking Strategy. Inspired by previous work (Fu et al., 2025), we also leverage the perplexity of LLMs to perform memory masking. Since high perplexity usually implies that LLMs are not confident of the generated content, answers with high perplexity may contain erroneous information (e.g., incoherent logic and hallucination). Thus, as an objective masking strategy, the perplexity of responses is measured, and half of the responses with high perplexity are filtered out.

Step 3. Reasoning with Masked Memories. With the new set of memories $\widehat{\mathcal{M}}_r^{(i)}$ obtained for each LLM agent through Eq. (2), in the $r+1$-th debate round, a set of new reasoning responses (memories) $\mathcal{M}_{r+1} = [A'_{\theta_1}(x^{\text{test}}, \widehat{\mathcal{M}}_r^{(1)}), A_{\theta_2}(x^{\text{test}}, \widehat{\mathcal{M}}_r^{(2)}), ..., A_{\theta_{N_\text{a}}}(x^{\text{test}}, \widehat{\mathcal{M}}_r^{(N_\text{a})})]$, where both correct and incorrect responses are included, is generated from agents with the query and the refined memories.

In our proposed MAD-M^2 framework, steps 2 and 3 are conducted iteratively until the end of the debate. At the final round of debate, majority voting is conducted to obtain the answer to the query.

3.2 More Discussions

In the previous section, we introduce a simple yet effective MAD-M^2 method to improve the robustness of the conventional multi-agent debate framework by reducing the incorrect memories derived in the previous debate round. In this section, we propose to provide more analyses and discussions about our proposed MAD-M^2 to have a more comprehensive understanding of its property.

Token Consumption Analysis. Due to the multi-round interactions in the multi-agent debate paradigm, the consumption of tokens of reasoning paradigms has become a significant concern in recent works (Liu et al., 2024; Zeng et al., 2025). In our proposed MAD-M^2, the introduction of the evaluation and masking step tends to result in more consumption of tokens. Thus, in order to comprehensively compare conventional MAD and our proposed MAD-M^2 frameworks, we follow Liu et al. (2024) to conduct an analysis on the token consumption of both multi-agent debate paradigms.

Let’s denote the token of the query x^{test} as T^{q}, the token of the output of the agent A_{θ_i} at the r-th debate round as $T^{\text{o}}_{i,r}$. Then, we formulate the token consumption of both MAD and MAD-M^{2}as:

$$N^{\text{token}}_{\text{MAD}} = N_{\text{a}} N_{\text{round}} T^{\text{q}} + N_{\text{a}} \sum_{r=2}^{N_{\text{round}}} \sum_{i=1}^{N_{\text{a}}} T^{\text{o}}_{r-1,i} + \sum_{r=1}^{N_{\text{round}}} \sum_{i=1}^{N_{\text{a}}} T^{\text{o}}_{r,i},$$

$$N^{\text{token}}_{\text{MAD}-\text{M}^2} \leq N_{\text{a}} N_{\text{round}} T^{\text{q}} + 2N_{\text{a}} \sum_{r=2}^{N_{\text{round}}} \sum_{i=1}^{N_{\text{a}}} T^{\text{o}}_{r-1,i} + \sum_{r=1}^{N_{\text{round}}} \sum_{i=1}^{N_{\text{a}}} T^{\text{o}}_{r,i}.$$

According to the results above, compared to the conventional MAD method, our proposed MAD-M^2 consumes a similar number of tokens. Specifically, even in the worst case, where all memories are preserved, MAD-M^2 consumes $N_{\text{a}} \sum_{r=2}^{N_{\text{round}}} \sum_{i=1}^{N_{\text{a}}} T^{\text{o}}_{r-1,i}$ more input tokens than MAD framework.

Comparison to Existing Works. In literature, many works have been done to perform memory selection in the multi-agent debate framework (Liu et al., 2024; Li et al., 2024; Zeng et al., 2025). Specifically, Li et al. (2024) proposes Sparse MAD (S-MAD) to formulate the MAD framework as a graph, where one agent can access the response of the other if the two agents are connected. Moreover, Zeng et al. (2025) proposes Selective Sparse MAD (S^2-MAD) to reduce the exchange of less informative memories and unproductive discussions among agents. Among these works, the most related work to our proposed MAD-M^2 is S-MAD. The main idea of S-MAD is that only static (predefined) topologies are taken into consideration for sparse communications among agents. Although dynamic topology is also mentioned in the paper, the memories are selected in a random way. However, in our proposed MAD-M^2, the selected memories from the previous round and the sparsity of communications are determined by LLM agents with a cost of more token consumption.

4 Experiments

In this section, in order to evaluate our proposed MAD-M^2 method, we propose to conduct experiments on both mathematical reasoning and language understanding benchmarks. Specifically, we first

Table 1: Empirical results of accuracy and token consumption. We evaluate four mainstream open-source reasoning LLMs on four mathematical reasoning and language understanding benchmarks.

Methods	**AIME24**		**AIME25**		**MMLU_Pro**		**MATH**	
	ACC (%) ↑	Tokens ↓	ACC (%) ↑	Tokens ↓	ACC (%) ↑	Tokens ↓	ACC (%) ↑	Tokens ↓
Qwen2.5-7B-Instruct								
CoT	6.7	×0.17	3.3	×0.17	40.6±6.3	×0.16	47.2±4.0	×0.17
CoT-SC	10.0	×1.01	**6.7**	×0.97	**42.8±3.9**	×1.00	**54.6±** 2.7	×1.02
MAD	10.0	×1.00	3.3	×1.00	42.0±6.3	×1.00	47.4± 3.6	×1.00
MAD-M^2(S)	**13.3**	×1.59	**6.7**	×1.56	41.0±3.0	×1.58	48.8±5.0	×1.64
MAD-M^2(O)	10.0	×1.00	3.3	×1.00	42.0±6.3	×1.00	47.8±3.3	×1.00
Qwen2.5-Math-7B-Instruct								
CoT	6.7	×0.15	10.0	×0.16	27.6±4.5	×0.17	59.6±3.8	×0.17
CoT-SC	13.3	×0.96	10.0	×0.98	**31.0±2.8**	×1.00	**63.4±4.8**	×0.97
MAD	13.3	×1.00	10.0	×1.00	27.2±1.3	×1.00	61.2±5.2	×1.00
MAD-M^2(S)	13.3	×1.50	10.0	×1.64	28.6±3.6	×1.69	61.4±5.5	×1.96
MAD-M^2(O)	13.3	×1.00	10.0	×1.00	27.2±1.3	×1.00	61.2±5.2	×1.00
DeepSeek-Math-7B-Instruct								
CoT	0.0	×0.13	0.0	0.16	**20.2±2.9**	×0.17	23.0±2.4	×0.17
CoT-SC	0.0	×0.99	0.0	×1.00	18.2±4.5	×0.99	**32.4±4.8**	×0.99
MAD	0.0	×1.00	0.0	×1.00	17.6±2.8	×1.00	26.4±4.4	×1.00
MAD-M^2(S)	0.0	×1.72	**3.3**	×1.72	17.4±4.3	×1.66	27.0±3.5	×1.74
MAD-M^2(O)	0.0	×1.00	0.0	×1.00	17.6±2.8	×1.00	26.4±4.4	×1.00
QwQ-32B								
CoT	73.3	×0.17	56.7	×0.17	74.0±6.3	×0.17	74.4±4.0	×0.17
CoT-SC	76.7	×1.02	**73.3**	×0.99	74.8±6.0	×1.01	**83.6±6.4**	×1.00
MAD	73.3	×1.00	70.0	×1.00	75.2±5.3	×1.00	78.8±4.5	×1.00
MAD-M^2(S)	**80.0**	×1.69	**73.3**	×1.63	75.2±4.8	×1.62	76.8±4.8	×1.61
MAD-M^2(O)	73.3	×1.03	**73.3**	×0.99	**76.0±3.7**	×1.00	77.4±5.0	×1.01

introduce the settings adopted in experiments and then provide the empirical results to validate the performance of MAD-M^2. Moreover, to obtain a comprehensive understanding of MAD-M^2, further empirical analyses are also conducted. Complete experimental settings are available in Appendix D.

4.1 Experimental Setups

Models. In this work, to validate the performance of MAD-M^2, we mainly consider 4 mainstream open-source large language models: Qwen2.5-7B-Instruct (Yang et al., 2024a), Qwen2.5-Math-7B-Instruct (Yang et al., 2024b), DeepSeek-Math-7B-Instruct (Shao et al., 2024), and QwQ-32B (Team, 2025). Specifically, we consider Qwen2.5-7B-Instruct as an LLM with weak capability, Qwen2.5-Math-7B-Instruct, DeepSeek-Math-7B-Instruct, and QwQ-32B as LLMs with powerful capability.

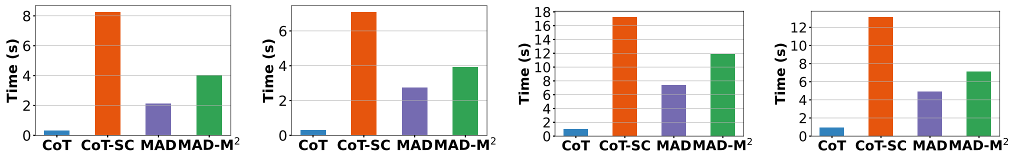
Benchmarks. We evaluate MAD-M^2 on both math reasoning and language understanding tasks. Specifically, we adopted MATH (Hendrycks et al., 2021), MMLU_Pro (Wang et al., 2024), AIME24, and AIME25 datasets in our experiments. Specifically, both MATH and MMLU_Pro represent the easy problem reasoning tasks, while AIME24 and AIME25 are the hard problem reasoning tasks.

Baselines. To validate the effectiveness of our proposed MAD-M^2 method, the following reasoning frameworks are adopted as baselines: (1) Chain-of-Thought (CoT) (Wei et al., 2022); (2) Self-Consistency Chain-of-Thoughts (CoT-SC) (Wang et al., 2023) with 7 independent reasoning paths; and (3) Multi-Agent Debate (MAD) (Du et al., 2023). For all the multi-agent debate frameworks above, the number of LLM agents and the number of debate rounds are set to 3 and 2, respectively.

4.2 Main Results

In this section, we evaluate our proposed MAD-M^{2}with both the subjective masking strategy (MAD-M^2(S)) and the objective masking strategy (MAD-M^2(O)) on the four mainstream mathematical and language understanding benchmarks. The accuracy and token consumption are reported in Table 1.

***Observation 1.* MAD-M^{2}outperforms MAD in most cases.** As shown in Table 1, we can observe that MAD-M^{2}achieves better performance than MAD in most cases. For example, with Qwen2.5-7B-



(a) MATH (b) MMLU_Pro (c) AIME24 (d) AIME25

Figure 3: Comparison of time consumption between MAD-M^{2}and baselines on Qwen2.5-7B-Instruct.

Table 2: Results of LLMs successfully identify and mask the erroneous memories.

Models	Rules	MATH	MMLU_Pro	AIME24	AIME25
Qwen2.5-7B-Instruct	Strict	32.0%	7.4%	90.0%	96.7%
	Loose	43.8%	33.4%	90.0%	96.7%
Qwen2.5-Math-7B-Instruct	Strict	54.4%	4.2%	100.0%	100.0%
	Loose	64.6%	4.4%	100.0%	100.0%
DeepSeek-Math-7B	Strict	7.8%	0.2%	96.7%	100.0%
	Loose	11.0%	2.2%	96.7%	100.0%
QwQ-32B	Strict	49.8%	15.6%	90.0%	86.7%
	Loose	52.0%	45.9%	90.0%	86.7%

Instruct, MAD-M^{2}achieves 3.3%, 3.4%, and 1.6% improvements on AIME24, AIME25, and MATH benchmarks, respectively. Moreover, with QwQ-32B, MAD-M^{2}also achieves 6.7%, 3.3%, and 0.8% improvements on AIME24 AIME25, and MMLU_Pro benchmarks, respectively. All these empirical results demonstrate the effectiveness of our proposed MAD-M^2.

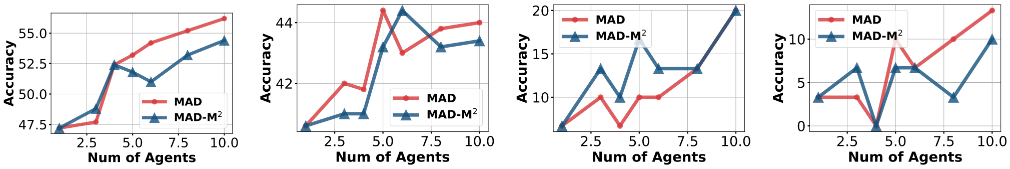
Observation 2. **MAD-M^{2}fails to outperform CoT-SC in almost all cases.** According to the results, we can observe that MAD-M^{2}consistently underperforms CoT-SC in most cases. This phenomenon is consistent with our theoretical results, which find that the performance of MAD can hardly outperform CoT-SC, in Section 2.2.1. This is also consistent with the results reported by Huang et al. (2023).

Observation 3. **Powerful LLMs achieve better performance on HPR, while weak LLMs perform better on EPR.** According to the empirical results, we find that weak LLMs, such as Qwen2.5-7B-Instruct, achieve more improvements on easy benchmarks (e.g., MATH and MMLU_Pro), while powerful LLMs, such as QwQ-32B, perform better on hard benchmarks (e.g., AIME). We conjecture that the reason for such a phenomenon is that the performance of powerful LLMs has reached a saturation point, where less incoherent content is included in responses, on easy benchmarks. In contrast, for weak LLMs, incoherent content remains in the responses even for easy benchmarks.

Observation 4. **Subjective masking strategy achieves better performance than objective masking strategy.** Empirically, according to the reported results, MAD-M^{2}with the subjective masking strategy outperforms MAD-M^{2}with the objective masking strategy in most cases. The reason for this phenomenon may be that the objective masking strategy implicitly assumes that the performance is positively related to the perplexity of LLMs. Nevertheless, MAD-M^{2}with the objective masking strategy still achieves comparable or even better performance than MAD/MAD-M^{2}in some cases.

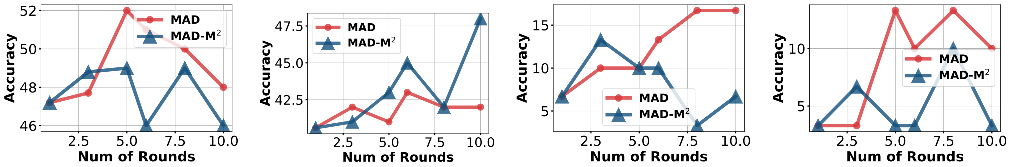
4.3 Analyses

Erroneous Memory Identification. The main goal of our proposed MAD-M^{2}is to identify the erroneous memories in the previous debate round and mask them for better performance. To validate the effectiveness of MAD-M^2, we propose to examine whether MAD-M^{2}correctly detects and masks the erroneous memories. Two rules are considered here: (1) *Strict rule*: All erroneous memories are detected and masked. (2) *Loose rule*: At least one erroneous memory is detected and masked. The results are reported in Table 2. According to the results, we can obtain the following observations. (1) LLMs can hardly identify all erroneous memories in most cases. Although all erroneous memories are identified in some cases of AIME benchmarks, we conjecture the reason here is that these memories have evident logical errors that can be easily detected since the problems are too hard. (2) Although erroneous memories can be identified on hard problems, the performance is not improved for those 7B LLMs (cf. Table 1). This phenomenon indicates that the capability of LLMs plays an essential role in improving the reasoning performance. (3) LLMs that are specific to math reasoning tasks



(a) MATH (b) MMLU_Pro (c) AIME24 (d) AIME25

Figure 4: Comparison of scaling of agents between MAD and MAD-M^2 on Qwen2.5-7B-Instruct.



(a) MATH (b) MMLU_Pro (c) AIME24 (d) AIME25

Figure 5: Comparison of debate round scaling between MAD and MAD-M^2 on Qwen2.5-7B-Instruct.

(e.g., Qwen-Math and DeepSeek-Math) can hardly identify the erroneous memories in the previous debate round, which helps explain why these LLMs perform poorly on language understanding tasks.

Token Consumption. In Section 3.2, we have theoretically demonstrated that our proposed MAD-M^2 consumes more tokens than MAD. Here, we compare the token consumptions of all methods listed in Table 1. According to the table, we find that multi-round/multi-agent reasoning consumes more tokens than the single-round/single-agent counterpart (e.g., CoT vs. CoT-SC & MAD vs. MAD-M^2). Specifically, MAD-M^{2}consumes about 60% more tokens than MAD. According to our empirical results, the high cost of tokens mainly results from input tokens, depending on LLMs. Specifically, for LLMs except QwQ-32B, the input tokens consumed by MAD-M^{2}are about $2 \sim 4$ times as large as MAD. For QwQ-32B, the consumption of input tokens is about $10 \sim 20$ times as large as MAD.

Time Consumption. To validate the efficiency, in Fig. 3, we compare the time consumption of MAD-M^{2}and other baselines on Qwen2.5-7B-Instruct. According to the visualization results, we find that MAD-M^{2}consumes more time than MAD but less time than CoT-SC. Meanwhile, we also notice that time consumption on solving hard problems is much more than that on easy problems.

Scaling of Agents. To study the scaling capability of MAD-M^2 on the number of agents, we propose to fix the number of debate rounds to 2 and increase the number of agents to 4, 5, 6, 8, and 10, respectively. The performance of Qwen2.5-7B-Instruct on various numbers of agents is visualized in Fig. 4. According to the figures, we can observe that both MAD and MAD-M^{2}benefit from the increase in agent numbers. Compared to MAD, MAD-M^{2}can achieve better performance when the number of agents is small (e.g., in MATH, AIME24, and AIME25 benchmarks).

Scaling of Number of Rounds. To study the scaling capability of our proposed MAD-M^2 on the number of debate rounds, we propose to fix the number of agents to 3 and increase the number of debate rounds to 5, 6, 8, and 10, respectively. The empirical results are visualized in Fig. 5. Different from increasing the number of agents, increasing the number of debate rounds does not increase the number of reasoning responses. According to our empirical results, we can observe that increasing the number of debate rounds fails to improve the reasoning performance. Instead, in most cases, we can observe a deterioration of performance.

5 Conclusion

Our work mainly focuses on the robustness of the conventional multi-agent debate framework and the effect of erroneous memories on LLM agents during the debate phase. In this paper, we first investigate the effect of erroneous memories in the MAD and find that LLM agents may be misled by those erroneous memories in the previous debate round and, in turn, generate incorrect reasoning responses in the next debate round. Moreover, we further theoretically demonstrate that the reasoning capability of LLM agents in the next debate round is closely related to the memories in the previous round, and enhancing such a capability can improve the performance of the multi-agent debate framework. Inspired by this, we propose a multi-agent debate framework, multi-agent debate with memory masking, to enhance the capability of agents by allowing them to mask incorrect memories in the previous round. Extensive empirical results demonstrate the efficacy of our proposed method.

ETHICS STATEMENT

Our work does not involve such concerns that should be claimed here.

REPRODUCIBILITY STATEMENT

In this paper, both theoretical and empirical results are included. For theoretical results, the necessary assumptions and the completed proofs have been provided in the main paper and the appendix, respectively. For the reproducibility of our experiments, we have provided detailed instructions of our experimental settings and the necessary introduction to the datasets in our paper.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. In Second Conference on Language Modeling, 2025.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In ICLR, 2024.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In AAAI, 2024.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. arXiv preprint arXiv:2407.21787, 2024.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In ICML, 2023.

Siqi Fan, Peng Han, Shuo Shang, Yequan Wang, and Aixin Sun. Cothink: Token-efficient reasoning via instruct models guiding reasoning models. arXiv preprint arXiv:2505.22017, 2025.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on rag meeting llms: Towards retrieval-augmented large language models. In Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining, pp. 6491–6501, 2024.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. arXiv preprint arXiv:2508.15260, 2025.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997, 2(1), 2023.

Shirley Anugrah Hayati, Minhwa Lee, Dheeraj Rajagopal, and Dongyeop Kang. How far can we extract diverse perspectives from large language models? arXiv preprint arXiv:2311.09799, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. arXiv preprint arXiv:2009.03300, 2020.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874, 2021.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. arXiv preprint arXiv:2310.01798, 2023.

Ziqi Jin and Wei Lu. Tab-cot: Zero-shot tabular chain of thought. arXiv preprint arXiv:2305.17812, 2023.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In NeurIPS, 2022.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. NeurIPS, 2020.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. Improving multi-agent debate with sparse communication topology. arXiv preprint arXiv:2406.11776, 2024.

Jinqing Lian, Xinyi Liu, Yingxia Shao, Yang Dong, Ming Wang, Zhang Wei, Tianqi Wan, Ming Dong, and Hailin Yan. Chatbi: Towards natural language to complex business intelligence sql. arXiv preprint arXiv:2405.00527, 2024.

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. arXiv preprint arXiv:2409.14051, 2024.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. arXiv preprint arXiv:2501.12570, 2025.

LINHAO LUO, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. In ICLR, 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 NeurIPS, 2023.

Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, et al. A survey of context engineering for large language models. arXiv preprint arXiv:2507.13334, 2025.

Marvin Minsky. The society of mind. Simon & Schuster, Inc., USA, 1986. ISBN 0671607405.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. arXiv preprint arXiv:2506.13131, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. NeurIPS, 2022.

Arjun Panickssery, Samuel Bowman, and Shi Feng. Llm evaluators recognize and favor their own generations. NeurIPS, 2024.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In ICLR, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. Dast: Difficulty-adaptive slow-thinking for large reasoning models. arXiv preprint arXiv:2503.04472, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In NeurIPS, 2023.

KaShun Shum, Shizhe Diao, and Tong Zhang. Automatic prompt augmentation and selection with chain-of-thought from labeled data. arXiv preprint arXiv:2302.12822, 2023.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers. In ICLR, 2025. URL https://openreview.net/forum?id=M23dTGWCZy.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. arXiv preprint arXiv:2408.03314, 2024.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In ICLR, 2024.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. Confidence improves self-consistency in llms. arXiv preprint arXiv:2502.06233, 2025.

Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, March 2025. URL https://qwenlm.github.io/blog/qwq-32b/.

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. In ICLR, 2022.

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. In ICLR, 2023.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. NeurIPS, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In NeurIPS, 2022.

Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Yang Wang. Pride and prejudice: Llm amplifies self-bias in self-refinement. ACL, 2024.

Yuchen Yan, Yongliang Shen, Yang Liu, Jin Jiang, Mengdi Zhang, Jian Shao, and Yueting Zhuang. Inftythink: Breaking the length limits of long-context reasoning in large language models. arXiv preprint arXiv:2503.06692, 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, Tianyi Tang, 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, 2024a.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. arXiv preprint arXiv:2409.12122, 2024b.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In NeurIPS, 2023.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. In NeurIPS, 2022.

Yuting Zeng, Weizhe Huang, Lei Jiang, Tongxuan Liu, Xitai Jin, Chen Tianying Tiana, Jing Li, and Xiaohua Xu. S^2-mad: Breaking the token barrier to enhance multi-agent debate efficiency. In NAACL, 2025.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In ICLR, 2023.

A Detailed Related Work

Recently, more and more attention has been attracted to large language models (Achiam et al., 2023; **?**; Yang et al., 2024a), which are parameterized with millions of weights and trained on numerous corpora. Along with the improvements in performance on conventional natural language tasks, one important emergent ability of these large language models is performing reasoning tasks.

LLM Reasoning. As the most impressive emergent capability of large language models, reasoning has received more and more attention from the community. Such a capability enables LLMs to address complex questions in a logical way as human beings based on their prior knowledge without further training. So far, numerous research works have been conducted to improve the reasoning capability of LLMs. One typical framework to enhance the reasoning capability of LLMs is Chain-of-Thought (CoT) (Wei et al., 2022). In CoT, LLMs address the questions by breaking the tasks into sequential steps. Based on CoT, some other variants are proposed to further improve the reasoning capability of LLMs (Zelikman et al., 2022; Wang et al., 2022; Shum et al., 2023). Among these works, a famous work is Self-Consistency Chain-of-Thought (CoT-SC) (Wang et al., 2022). CoT-SC improves the reasoning capability of LLMs significantly by simply sampling multiple reasoning trajectories and performing majority voting with these trajectories. In addition to modeling the reasoning trajectories into the chain of thoughts, some works also try to explore other structures. Specifically, Jin & Lu (2023) (Tab-CoT) proposes to model the reasoning trajectories in a highly structured way. Yao et al. (2023) proposes to model the reasoning trajectories into a tree-like path (a.k.a Tree-of-Thought, ToT). This further facilitates evaluating multiple reasoning paths and self-assessment. However, both chain-based or tree-based reasoning paradigms are constrained to linear reasoning tasks. To solve this problem, Graph-of-Thoughts (Besta et al., 2024) is proposed to model the reasoning into a flexible graph. In such a way, the non-linear tasks can be well solved.

Multi-agent Debate. Multi-agent debate (MAD) (Du et al., 2023) can be treated as a scaling of conventional reasoning paradigms, such as CoT. The main goal of MAD is to perform reasoning with multiple LLMs in the way of debate as human beings. Specifically, in such a case, the reasoning is performed with multiple LLMs in several rounds, and the final answer is generated via majority voting. By taking previous responses into consideration, the answers generated in the new round can be further refined. Compared to the conventional reasoning paradigms, such as CoT, MAD consumes more resources (e.g., tokens and time) (Du et al., 2023; Li et al., 2024; Liu et al., 2024; Zeng et al., 2025). Moreover, due to the inconsistency of the quality, multi-agent debate also suffers from the noisy responses (Zeng et al., 2025). Thus, a series of work is proposed to reduce the resource consumption and improve the performance of MAD by polishing the memories. Specifically, Sparse MAD (S-MAD) (Li et al., 2024) proposes to formulate the MAD framework as a graph and sparsify the topology of the graph, which is able to simultaneously reduce resource consumption and improve the performance. Moreover, Liu et al. (2024) proposes Group Debate (GD) to divide all agents into several debate groups and only allow agents to debate in their own group. Further, Selective Sparse MAD (S^2-MAD (Zeng et al., 2025)) is proposed to reduce the redundant content and inproductive discussions in debate to improve the efficiency and performance of MAD.

Context Engineering. Context Engineering methods (Mei et al., 2025) manipulate unstructured streaming LLM reasoning to construct structured context systems, enabling sophisticated LLM-driven MAD systems and agentic systems by managing scaling context length, where numerous tokens are irrelevant or confusing for answers. Recent advancements in context engineering are two-fold: context augmentation and context compression. Context augmentation methods extend critical information in context to optimize reasoning performance. Retrieval-augmented Generation (RAG) methods (Lewis et al., 2020; Gao et al., 2023; Fan et al., 2024) provide access to external information sources, including databases (Lian et al., 2024), knowledge graphs (Sun et al., 2024; LUO et al., 2024), and textual document collections (Asai et al., 2024; Sarthi et al., 2024), enabling access to relevant knowledge. Additionally, comprehensive feedback approaches like Self-Refine (Madaan et al., 2023) employ LLMs to evaluate their prior reasoning for self-improvement, while Reflexion (Shinn et al., 2023), STaR (Zelikman et al., 2022), and AlphaEvolve (Novikov et al., 2025) introduce reliable external environment feedback for consistent performance improvement. Context compression methods address overlong context issues that challenge finite context window sizes by dynamically reducing reasoning context. INFTYTHINK (Yan et al., 2025) iteratively summarizes prior reasoning contexts for short-context reasoning, while CoThink (Fan et al., 2025) employs short-response LLMs to orchestrate reasoning strategy and dynamically control context length. Additionally, O1-

Pruner (Luo et al., 2025), L1 (Aggarwal & Welleck, 2025), and DAST (Shen et al., 2025) consider the reasoning length during training to encourage LLMs to solve problems with fewer tokens.

B Proofs

B.1 Proof of Proposition 2.2

Proof. Since CoT-SC performs reasoning on the given query by voting the majority on a set of independently generated multiple responses, given Assumption 2.1, the number of the correct responses conforms to a binomial distribution $N_{\text{cor}} \sim \mathcal{B}(N_{\text{sc}}, k)$:

$$P(N_{\text{cor}} = k) = \binom{N_{\text{sc}}}{k} p^k (1-p)^{N_{\text{sc}}-k},$$

where $\mathcal{B}$ denotes the binomial distribution. Thus, the case that CoT-SC correctly answers the question can be formulated as the event: $N_{\text{cor}} > \frac{N_{\text{sc}}}{2}$. Then, we further have the probability that CoT-SC correctly infers the answer to the query

$$\begin{aligned} P(N_{\text{cor}} > \frac{N_{\text{sc}}}{2}) &= \sum_{k=\lceil \frac{N_{\text{sc}}}{2}+1 \rceil}^{N_{\text{sc}}} P(N_{\text{cor}} = k) \\ &= \sum_{k=\lceil \frac{N_{\text{sc}}}{2}+1 \rceil}^{N_{\text{sc}}} \binom{N_{\text{sc}}}{k} p^k (1-p)^{N_{\text{sc}}-k}, \end{aligned}$$

where $\lceil \cdot \rceil$ is the ceiling operator, and $\binom{N_{\text{sc}}}{k} = \frac{N_{\text{sc}}!}{k!(N_{\text{sc}}-k)!}$ is the binomial coefficient with N_{sc} and k.

Consider a set of independent random variables $\{X_i\}_{i=1}^{N_{\text{sc}}}$, where $X_i \in \{0, 1\}$, $P(X_i = 1) = p$ and $P(X_i = 0) = 1 - p$. Let $N_{\text{cor}} = \sum_{i=1}^{N_{\text{sc}}} X_i$, we then have $E[X_i] = p$ and $E[N_{\text{cor}}] = N_{\text{sc}}p$. With the Hoeffding Inequality, we have

$$P(N_{\text{cor}} - E[N_{\text{cor}}] \geq t]) \leq \exp\left(-\frac{2t^2}{N_{\text{sc}}}\right),$$

If $p < \frac{1}{2}$, then $E[N_{\text{cor}}] = N_{\text{sc}}p < \frac{N_{\text{sc}}}{2}$. Let $t = (\frac{1}{2} - p)N_{N_{\text{sc}}} > 0$:

$$P(N_{\text{cor}} > \frac{N_{\text{sc}}}{2}) \leq \exp\left(-2N_{\text{sc}}\left(\frac{1}{2} - p\right)^2\right).$$

Else, if $p \geq \frac{1}{2}$, then $E[N_{\text{cor}}] \geq \frac{N_{\text{sc}}}{2}$. Thus, we consider the complementary case: $P(N_{\text{cor}} - E[N_{\text{cor}}] \leq \frac{N_{\text{sc}}}{2})$. Then, let $t = (p - \frac{1}{2})N_{\text{sc}} > 0$, we can get:

$$P(N_{\text{cor}} \leq \frac{N_{\text{sc}}}{2}) \leq \exp\left(-2N_{\text{sc}}\left(\frac{1}{2} - p\right))^2\right).$$

Thus, for the original case $P(N_{\text{cor}} > \frac{N_{\text{sc}}}{2})$, we have

$$P(N_{\text{cor}} > \frac{N_{\text{sc}}}{2}) \geq 1 - \exp\left(-2N_{\text{sc}}\left(\frac{1}{2} - p\right)^2\right).$$

The proof is completed. □

B.2 Proof of Proposition 2.3

Proof. In a 2-round multi-agent debate case, where N_{a} agents are involved in each debate round, we first consider the initial debate round, where the responses are only conditioned on the query prompt. Thus, the probability of MAD achieving $N_{\text{cor}}^{(1)} \in \{0, 1, 2, ..., N_{\text{a}}\}$ correct reasoning responses is:

$$P(N_{\text{cor}}^{(1)} = j) = \binom{N_{\text{a}}}{j} p^j (1-p)^{N_{\text{a}}-j}, \text{where } j \in \{0, 1, 2, ..., N_{\text{a}}\}.$$

For the second round, assume that the probability of LLM agents correctly reasoning the answer is modified to $e^{-N_{\mathrm{e}}}$, where $N_{\mathrm{e}} = N_{\mathrm{a}} - N_{\mathrm{cor}}^{(1)}$, depending on the number of correct memories in the previous debate round. We consider that new responses are generated independently in the second round, and the number of correct responses in the second round also conforms to a binomial distribution $N_{\mathrm{cor}}^{(2)} \sim \mathcal{B}(N_{\mathrm{a}}, e^{j-N_{\mathrm{a}}})$. In this case, the probability that the answer is correctly inferred is:

$$\begin{aligned} P(N_{\mathrm{cor}}^{(2)} > \frac{N_{\mathrm{a}}}{2}) &= \sum_{j=0}^{N_{\mathrm{a}}} \sum_{k=\lceil \frac{N_{\mathrm{a}}}{2}+1 \rceil}^{N_{\mathrm{a}}} P(N_{\mathrm{cor}}^{(2)} = k | N_{\mathrm{cor}}^{(1)} = j) \\ &= \sum_{j=0}^{N_{\mathrm{a}}} \sum_{k=\lceil \frac{N_{\mathrm{a}}}{2}+1 \rceil}^{N_{\mathrm{a}}} P(N_{\mathrm{cor}}^{(2)} = k) P(N_{\mathrm{cor}}^{(1)} = j) \\ &= \sum_{j=0}^{N_{\mathrm{a}}} P(N_{\mathrm{cor}}^{(1)} = j) \sum_{k=\lceil \frac{N_{\mathrm{a}}}{2}+1 \rceil}^{N_{\mathrm{a}}} \binom{N_{\mathrm{a}}}{k} e^{k(j-N_{\mathrm{a}})} (1 - e^{j-N_{\mathrm{a}}})^{N_{\mathrm{a}}-k}, \end{aligned}$$

where $P(N_{\mathrm{cor}}^{(1)} = j) = \binom{N_{\mathrm{a}}}{j} p^j (1-p)^{N_{\mathrm{a}}-j}$.

Then, similar to Proposition 2.2, if $e^{j-N_{\mathrm{a}}} < \frac{1}{2}$, we have the upper bound:

$$P(N_{\mathrm{cor}}^{(2)} > \frac{N_{\mathrm{a}}}{2}) \leq \sum_{j=0}^{N_{\mathrm{a}}} \binom{N_{\mathrm{a}}}{j} p^j (1-p)^{N_{\mathrm{a}}-j} \exp\left(-2N_{\mathrm{a}} \left(\frac{1}{2} - e^{j-N_{\mathrm{a}}} \right)^2 \right).$$

Else, if $e^{j-N_{\mathrm{a}}} \geq \frac{1}{2}$, we have the lower bound:

$$P(N_{\mathrm{cor}}^{(2)} > \frac{N_{\mathrm{a}}}{2}) \geq \sum_{j=0}^{N_{\mathrm{a}}} \binom{N_{\mathrm{a}}}{j} p^j (1-p)^{N_{\mathrm{a}}-j} \left(1 - \exp\left(-2N_{\mathrm{a}} \left(\frac{1}{2} - e^{j-N_{\mathrm{a}}} \right)^2 \right) \right).$$

The proof is completed. □

C Detailed Analysis on Token Consumption

Denote the token of the query x^{test} as T^{q}, the token of the output of the agent A_{θ_i} at the r-th debate round as $T_{i,r}^{\mathrm{o}}$. Then, we formulate the token consumption of both MAD and MAD-M^{2}as follows.

For MAD, at the initial round, the query x^{test} is fed into N_{a} LLM agents and then N_{a} responses are generated from these LLM agents. Thus, the consumption of MAD at the initial debate round is:

$$N_1^{\mathrm{token}} = N_{\mathrm{a}} T^{\mathrm{q}} + \sum_{i=1}^{N_{\mathrm{a}}} T_{1,i}^{\mathrm{o}}.$$

Then, from the second debate round, each agent will take all N_{a} outputs in the last debate round and output a new reasoning response with the query. Thus, the consumption of tokens for MAD at the r-th debate round is:

$$N_r^{\mathrm{token}} = N_{\mathrm{a}} \left(T^{\mathrm{q}} + \sum_{i=1}^{N_{\mathrm{a}}} T_{r-1,i}^{\mathrm{o}} \right) + \sum_{i=1}^{N_{\mathrm{a}}} T_{r,i}^{\mathrm{o}}.$$

Thus, the total consumption of tokens of MAD can be formulated as:

$$\begin{aligned} N_{\mathrm{MAD}}^{\mathrm{token}} &= \sum_{r=1}^{N_{\mathrm{round}}} N_r^{\mathrm{token}} \\ &= N_{\mathrm{a}} N_{\mathrm{round}} T^{\mathrm{q}} + N_{\mathrm{a}} \sum_{r=2}^{N_{\mathrm{round}}} \sum_{i=1}^{N_{\mathrm{a}}} T_{r-1,i}^{\mathrm{o}} + \sum_{r=1}^{N_{\mathrm{round}}} \sum_{i=1}^{N_{\mathrm{a}}} T_{r,i}^{\mathrm{o}}. \end{aligned}$$

For MAD-M^2, at the initial round, the consumption of tokens is the same as MAD:

$$N_1^{\text{token}} = N_{\text{a}} T^{\text{q}} + \sum_{i=1}^{N_{\text{a}}} T_{1,i}^{\text{o}}.$$

From the second round, the agents will first evaluate the previous memories and mask the potentially incorrect memories. In this step, all memories are fed into each agent, and the agent will output “yes”, “no”, or “unsure”. Here, we ignore the tokens of outputs and the instructions. We then formulate the consumption of tokens at this step as:

$$N_r^{\text{eval\_token}} = N_{\text{a}} \sum_{i=1}^{N_{\text{a}}} T_{r-1,i}^{\text{o}}.$$

Then, based on the selected memories $\widehat{\mathcal{M}}_{r-1}^{(i)}$, the input tokens for each agent in the r-th debate round should be no more than the sum of the query tokens and the tokens of all previous tokens. Thus, considering the output tokens, we have:

$$N_r^{\text{token}} \leq N_{\text{a}} \left(T^{\text{q}} + \sum_{i=1}^{N_{\text{a}}} T_{r-1,i}^{\text{o}} \right) + \sum_{i=1}^{N_{\text{a}}} T_{r,i}^{\text{o}}.$$

Thus, the total consumption of MAD-M^2 can be formulated as:

$$\begin{aligned} N_{\text{MAD-M}^2}^{\text{token}} &= N_1^{\text{token}} + \sum_{r=2}^{N_{\text{round}}} \left(N_r^{\text{eval\_token}} + N_r^{\text{token}} \right) \\ &\leq N_{\text{a}} T^{\text{q}} + \sum_{i=1}^{N_{\text{a}}} T_{1,i}^{\text{o}} + \sum_{r=2}^{N_{\text{round}}} \left(N_{\text{a}} \sum_{i=1}^{N_{\text{a}}} T_{r-1,i}^{\text{o}} + N_{\text{a}} \left(T^{\text{q}} + \sum_{i=1}^{N_{\text{a}}} T_{r-1,i}^{\text{o}} \right) + \sum_{i=1}^{N_{\text{a}}} T_{r,i}^{\text{o}} \right) \\ &= N_{\text{a}} N_{\text{round}} T^{\text{q}} + 2N_{\text{a}} \sum_{r=2}^{N_{\text{round}}} \sum_{i=1}^{N_{\text{a}}} T_{r-1,i}^{\text{o}} + \sum_{r=1}^{N_{\text{round}}} \sum_{i=1}^{N_{\text{a}}} T_{r,i}^{\text{o}}. \end{aligned}$$

D Complete Experimental Settings

In this section, we provide more detailed implementations adopted in the experiment section of our paper to ensure that the results reported in this paper are reproducible.

D.1 Test Data Settings

In this paper, all single-round and multi-round reasoning methods are evaluated on MATH, MMLU_Pro, AIME24, and AIME25 datasets. However, due to the multi-round paradigms of the multi-agent debate framework, it is really expensive to evaluate all the test data of MATH and MMLU_Pro. Thus, to avoid this problem, we follow previous works (Du et al., 2023) to perform evaluation on a subset of 100 randomly sampled test data.

D.2 Hyperparameter Settings

Hyperparameters of LLMs. In this paper, all reasoning responses are performed on LLMs with a temperature of 1.0 and a top p of 1.0.

Fairness of Comparison between CoT-SC and MAD. For the CoT-SC baseline, the final answer is achieved by performing majority voting among 5 responses that are independently generated from LLMs in the way of CoT. Since the MAD framework is performed with 3 agents in 2 rounds, the total number of reasoning responses in MAD is 6. Thus, we think the comparison between CoT-SC and MAD in this paper is fair.

Computational Resources. In this paper, all experiments are conducted through the platform API. Thus, none of other computational resources, such as GPUs are involved.

D.3 Details of Benchmarks

In this work, all baselines and our proposed MAD-M^2 are evaluated in four mainstream mathematical reasoning and language understanding benchmarks, which are AIME24, AIME25, MATH, and MMLU_Pro. In this section, we provide a brief introduction for each of these four benchmark.

- **AIME 2024:** A challenging competition mathematical question set on the 2024 American Invitational Mathematics Examination for high school students.
- **AIME 2025:** The questions of the American Invitational Mathematics Examination in 2025, providing the latest difficult mathematical problems.
- **MATH:** A comprehensive mathematical dataset designed to rigorously challenge large language models' reasoning abilities in algebra, geometry, number theory, and combinatorics, thoroughly evaluating their mathematical understanding and problem-solving capabilities.
- **MMLU-Pro:** MMLU-Pro is an advanced extension of the Massive Multitask Language Understanding (MMLU) dataset (Hendrycks et al., 2020), encompassing 57 diverse tasks across domains such as mathematics, computer science, biology, and physics. Compared to MMLU, MMLU-Pro introduces more complex, reasoning-intensive problems to enhance the evaluation of advanced language models.

D.4 Details of Prompts

CoT prompt for math problem (AIME24 AIME25 MATH)

Please solve this mathematical problem step by step, and provide the final answer in \boxed{} format.

CoT Prompt for MMLU-Pro

The following are multiple-choice questions (with answers) about {$}. Think step by step and then finish your answer with the answer is (X), where X is the correct letter choice.

Debate Prompt

These are the solutions to the problem from other agents:
{context} Use the opinions from other agents as additional information.
{question} Please think step by step and solve the problem.

Masking Prompt

Evaluate the given solutions based on the question. **Your response MUST end with the following format: <label>YES</label> or <label>NO</label> or <label>NOT SURE</label>.** Return YES if the solution is completely correct, NO if any part of the solution is incorrect, and NOT SURE if you are unsure.
{question}
{solution}