S^2-MAD: Breaking the Token Barrier to Enhance Multi-Agent Debate Efficiency

Anonymous ACL submission

Abstract

Large language models (LLMs) have demonstrated remarkable capabilities across various natural language processing (NLP) scenarios, but they still face challenges when handling complex arithmetic and logical reasoning tasks. While Chain-Of-Thought (CoT) reasoning, self-consistency (SC) and self-correction strategies have attempted to guide models in sequential, multi-step reasoning, Multi-agent Debate (MAD) has emerged as a viable approach for enhancing the reasoning capabilities of LLMs. By increasing both the number of agents and the frequency of debates, the performance of LLMs improves significantly. However, this strategy results in a significant increase in token costs, presenting a barrier to scalability. To address this challenge, we introduce a novel sparsification strategy designed to reduce token costs within MAD. This approach minimizes ineffective exchanges of information and unproductive discussions among agents, thereby enhancing the overall efficiency of the debate process. We conduct comparative experiments on multiple datasets across various models, demonstrating that our approach significantly reduces the token costs in MAD to a considerable extent. Specifically, compared to MAD, our approach achieves an impressive reduction of up to 94.5% in token costs while maintaining performance degradation below 2.0%.

1 Introduction

Large language models (LLMs) have shown exceptional capabilities across a variety of natural language processing (NLP) tasks (Achiam et al., 2023; Brown et al., 2020; Bubeck et al., 2023; Radford et al., 2018, 2019; Touvron et al., 2023a,b; Anil et al., 2023; Chowdhery et al., 2023). However, even the most advanced LLMs exhibit limitations in complex mathematical reasoning and logical inference scenarios (Liu et al., 2023).

To address these challenges, researchers have introduced techniques such as Chain-of-Thought (CoT) reasoning (Wei et al., 2022) which decomposes complex problems into sequential steps, and self-consistency (SC) mechanisms (Wang et al., 2022), along with self-correction strategies (Welleck et al., 2022; Madaan et al., 2024; Shinn et al., 2024).

Despite these innovations, studies have shown that LLMs still struggle to improve through self-correction alone (Huang et al., 2023; Valmeekam et al., 2023; Stechly et al., 2023). An emerging alternative is the Multi-agent Debate (MAD) framework, in which multiple independent agents propose and critique their own answers through rounds of debate, ultimately converging on a more robust consensus (Sun et al., 2023). MAD has demonstrated promise in addressing the limitations of LLM self-correction by leveraging diverse agent perspectives to refine answers over iterative discussions (Chan et al., 2023; Du et al., 2023; Liang et al., 2023). However, as the number of agents and debate rounds increase, the token cost escalates significantly, limiting the scalability of MAD, especially in resource-constrained environments (Li et al., 2024; Liu et al., 2024). To alleviate the token cost problem in multi-agent debates, researchers have proposed several strategies. For instance, (Du et al., 2023) summarizes agents' outputs at the end of each round, while (Sun et al., 2023) introduces a "forgetting" mechanism, where only the previous round's outputs is retained for future rounds. Another approach, Sparse-MAD (S-MAD) (Li et al., 2024), reduces communication overhead by limiting information exchange to neighboring agents. GroupDebate (GD) (Liu et al., 2024) further reduces token cost by clustering agents into smaller debate groups that exchange intermediate results between groups.

Although the reduction in token cost have achieved by the aforementioned approaches, our

Question: Bob is in charge of doing laundry for a large hotel. Each room has two sheets, one comforter, twice as many pillow cases as sheets and twice as many towels as pillow cases. How many pieces of laundry are there in 80 rooms?

Agent 1:
First, let's calculate how many of each item are in each room： 2 sheets, 1 comforter, 4 pillow cases (twice as many as sheets), 8 towels (twice as many as pillow cases). Therefore, in each room, there are a total of 2 + 1 + 4 + 8 = 15 pieces of laundry. For 80 rooms, the total number of pieces of laundry would be 15 * 80 = 1200.

Agent 2:
Let's break it down by each item of laundry per room： 2 sheets, 1 comforter, 2 pillow cases (twice as many as sheets), 4 towels (twice as many as pillow cases). Therefore, each room has a total of 2 + 1 + 2 + 4 = 9 pieces of laundry. For 80 rooms, the total number of pieces of laundry would be 80 rooms * 9 pieces of laundry per room = 720 pieces of laundry.

Agent 3:
Let's break down the number of each item in one room: 2 sheets, 1 comforter, 4 pillow cases (twice as many as sheets), 8 towels (twice as many as pillow cases). Adding all of these up, there are a total of 2 + 1 + 4 + 8 = 15 pieces of laundry in one room. Since there are 80 rooms, we can multiply the number of pieces of laundry in one room by the number of rooms: 15 pieces/room * 80 rooms = 1200 total pieces of laundry.

Figure 1: **Redundant Viewpoints Exchange between Agents.** The perspectives of Agent 1 and Agent 3 demonstrate a notable similarity. Throughout the debate, these viewpoints are exchanged with Agent 2, who receives these akin and repetitive viewpoints.

experiment reveals a substantial presence of redundancy and duplicate information in the inter-agent information exchange. As depicted in Figure 1, Agent 1 and Agent 3 exhibit repetitive viewpoints, leading to exacerbate the issue of token cost due to redundant duplication during the inter-agent information exchange. The issue of redundancy and duplication primarily stems from two potential factors: the limited solution space inherent in complex reasoning tasks, and the tendency of large language models to generate repetitive responses when faced with similar inputs (Holtzman et al., 2019; Xu et al., 2022; Yan et al., 2023).

To address these limitations, we propose a novel approach **S**elective **S**parse **M**ulti-**A**gent **D**ebate (S^2-MAD), as shown in Figure 2. This approach utilizes a Decision-Making Mechanism to determine whether to participate in the debate, thereby further reducing token costs within multi-agent debates. Specifically, based on a grouping strategy, S^2-MAD first generates initial viewpoints for the agents. In each round of debate, the Decision-Making Mechanism enables agents to selectively incorporate non-redundant responses that differ from their current viewpoints for answer checking and updating. The agents have the option to selectively engage in both intra-group and inter-group discussions, enabling them to actively participate in debates. The process concludes either when consensus is reached among the agents or when a final answer is obtained through majority voting.

To validate the effectiveness of S^2-MAD, we conduct a theoretical analysis of total token costs and perform extensive experiments across five tasks using different models. These experiments compare S^2-MAD with existing multi-agent debate strategy as well as single-agent reasoning approaches, demonstrating its capability to significantly reduce token counts while maintaining comparable accuracy. Specifically, S^2-MAD reduces token cost by up to 94.5% compared to MAD, 90.2% compared to MAD-Sparse, and 87.0% compared to GD, while also significantly reducing token cost by up to 81.7% compared to CoT-SC. Importantly, these reductions come with a performance degradation of less than 2.0%, demonstrating that S^2-MAD maintains high accuracy while minimizing communication overhead.

The main contributions of this paper are as follows:

1. We propose S^2-MAD, an innovative sparse multi-agent debate strategy with Dicision-Making Mechanism that reduces redundant information and inefficient debate.

2. We theoretically demonstrate the token cost advantages of S^2-MAD over MAD, S-MAD, and GD.

3. We validate the effectiveness of S^2-MAD across five datasets using commercial and open-source models, demonstrating a significant reduction in token costs with minimal performance loss.

2 Preliminary

Problem Definition. MAD, which integrates multiple agents for interactive communication to derive solutions, has demonstrated an effective

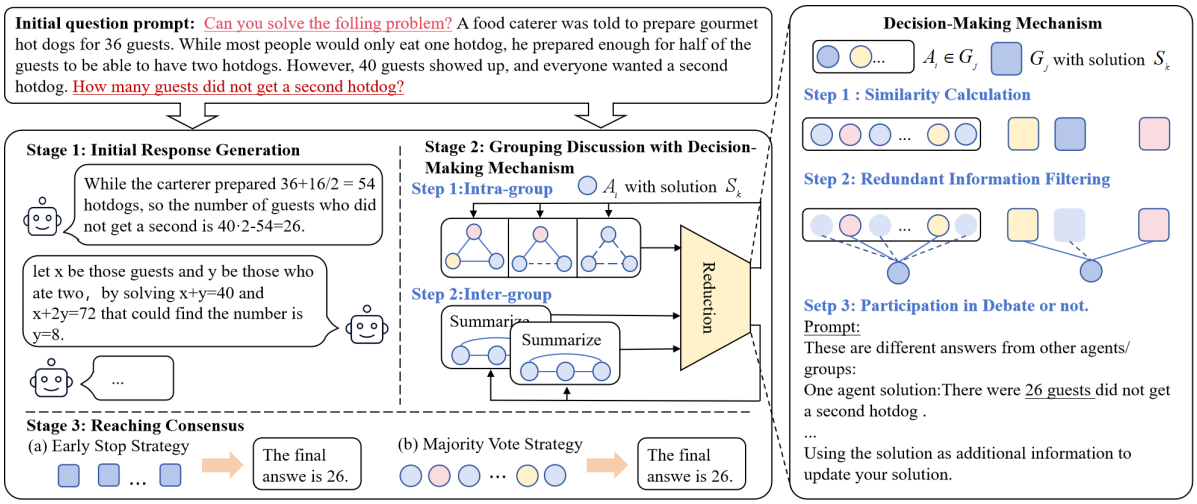

Figure 2: **Process of S^2-MAD.** The S^2-MAD includes three stages: all agents generate initial responses independently at the first round and participate in group discussions to reach consensus under a Decision-Making Mechanism, which comprises: (1) Similarity calculation module accesses the similarity of responses either between or within groups. (2) Redundancy filter module filters redundant information, retaining only unique information that differs from the agent's own perspective. (3) Conditional participation module decide to participate in debate or not.

approach in the application of LLMs, particularly in addressing complex logical reasoning and mathematical problems (Liang et al., 2023; Chan et al., 2023). Given an input question Q that requires an answer, a total of M participating Language Model (LLM)-based agents engage in a multi-round debate, which is denoted as A_i, where $i \in \{1, 2, \ldots, M\}$. Given a total of T debate rounds, each round of debate is denoted as $t \in \{1, 2, \ldots, T\}$. We define the output of each agent A_i at round t as O_i^t. We assume that the upper bound of the token cost for each agent's output is C. Our goal is to maximize answer accuracy while minimizing token consumption through optimizing the interaction patterns among the agents in multi-agent debate.

MAD-based Methods and Token Cost. (i) MAD (Liang et al., 2023) involves several steps. Initially, each agent is provided with a question and generates an individual response. These responses then form the new input context for each agent, leading to the generation of subsequent responses. This debate procedure is repeated over multiple rounds, with the final answer derived through majority voting. The token cost complexity is $Token^{MAD} = \mathcal{O}\left(MTQ + (M^2T + MT^2)C\right)$. (ii) S-MAD (Li et al., 2024) decreases token consumption by sparsifying the fully connected topology of information exchange among agents within the standard MAD framework. Let P_r denote the probability of each edge being removed. The token cost complexity can be represented as $Token^{S-MAD} = \mathcal{O}\left(MTQ + P_r M^2 TC\right)$. (iii) GD (Liu et al., 2024) reduces token consumption through a group discussion strategy. Let N denote the number of groups and S represent the number of inter-group debate stages. The token cost complexity is computed as $Token^{GD} = \mathcal{O}\left(MTQ + (\frac{M^2T}{N} + MSN)C\right)$.

3 Methodology

In this section, we first introduce the overall process of S^2-MAD along with the details of Decision-Making Mechanism. And then we provide mathematical analysis of the token cost for our method subsequently.

3.1 Selective Sparse MAD Process

As illustrated in Figure 2, the debate process of S^2-MAD consists of three main stages: the generation of initial responses, group discussions under the Decision-Making Mechanism and finally reaching consensus.

Initial Response Generation. In the initial round of debate, each agent is initialized as a LLM. To simulate diverse thought processes and ensure the generation of varied solutions, we employ a random decoding strategy by adjusting model temperatures. During the first round, all agents independently produce their respective solutions for the given problem.

Grouping Discussion with Decision-Making Mechanism. From the second round onward, the

Decision-Making Mechanism empowers agents to evaluate whether to engage in debate by assessing the similarity of intra- or inter-group viewpoints relative to their own perspectives. Agents will actively participate in debates when they encounter responses that present differing viewpoints, either from within their group or from other groups. Following these discussions, agents update their answers accordingly based on insights gained during the debate process.

Reaching Consensus. Our approach incorporates an early termination mechanism that allows us to conclude the debate when information has been exchanged between groups and all summarized viewpoints align. Conversely, if discrepancies remain among agents' solutions after the debate concludes, a majority vote will determine which solution is accepted as consensus.

3.2 Decision-Making Mechanism.

Upon receiving information, the decision-making mechanism first employs the Similarity Calculation Module to calculate dissimilarities among different pieces of information. Subsequently, it eliminates redundant perspectives of agents through the Redundancy Filtering Module. Finally, Conditional Participation Module is utilized to determine whether the agent should engage in the debate.

Similarity Calculation Module. Following the generation of outputs, each agent undertakes a comprehensive assessment of the similarity between its own outputs and those produced by other agents or groups. This evaluation can be conducted through various methodologies; in this context, we employ a straightforward approach that involves analyzing key points within the outputs to determine their degree of similarity. By focusing on essential elements, agents can effectively gauge how closely aligned their perspectives are with those presented by others.

Redundancy Filtering Module. Prior to engaging in the debate, agents systematically filter all incoming information to ensure relevance and uniqueness. Outputs that are identified as similar to either their own or previously received viewpoints are promptly discarded from consideration. This rigorous filtering process guarantees that each agent exclusively considers unique perspectives during discussions, thereby minimizing redundancy and fostering a more dynamic exchange of ideas.

Conditional Participation Module. Agents actively engage in debate when divergent viewpoints exist within or among groups, recognizing that such differences enrich the discourse and lead to more robust conclusions. Conversely, if all outputs align consistently without variation, agents will opt to remain silent rather than contribute redundant information. At the conclusion of each round of debate, agents update their knowledge base with accepted viewpoints gleaned from interactions with others; this iterative learning process enhances their ability to respond thoughtfully and effectively in subsequent rounds.

3.3 Token Cost Analysis

In S^2-MAD, we summarize the outputs from other groups at the end of each stage. Given a group of agents G_j which has completed a stage s of debate, we denote its summary as Sum_j^s. Since in S^2-MAD each agent determines participation based on whether viewpoints are consistent, we define the number of Agents with differing viewpoints from the ith Agent D_i is:

$$\begin{cases} \sum_{i' \in G_j} Sim(O_i^{t-1}, O_{i'}^{t-1}) < \epsilon, \\ \qquad (s-1)R+1 < t < min(sR, T) \\ \sum_{1}^{N} Sim(O_i^{t-1}, Sum_j^{s-1}) < \epsilon, \\ \qquad\qquad t = (s-1)R+1 \end{cases} \tag{1}$$

Therefore, apart from generating the initial answer, the probability of Agent A_i participating in the debate in round t is

$$P_i^t = \begin{cases} 1, & D_i^t > 0 \\ 0, & D_i^t = 0 \end{cases} \tag{2}$$

Then token cost $Token_s^t$ in round t at stage s is:

$$\sum_{j=1}^{N} \sum_{i \in G_j} P_i^t (Q + O_i^t + \frac{ND_i^t}{M} \sum_{i' \in G_j} O_{i'}^{t-1}) \tag{3}$$

where $(s-1)R+1 < t <= min(sR, T)$, and

$$\sum_{i=1}^{M} P_i^t (Q + O_i^{t-1} + O_i^t + \frac{D_i^t}{N} \sum_{j=1}^{N} Sum_j^{s-1}) \tag{4}$$

where $t = (s-1)R+1$. Finally, the total token cost of S^2-MAD is $Token =$

$\mathcal{O}\left(MTQ + (\frac{M^2T}{N} + MSN)CP\right)$, where C represents the upper bound on the token number for each agent's response and the generated summary, P represents the upper bound of the average probability of each Agent participating in the debate globally. More calculation details are shown in Appendix B.

Discussion. From the perspective of total token cost complexity comparison, S^2-MAD exhibits the same token cost complexity as standard MAD since the initial viewpoints of Agents are generated by retaining the question input. However, for the same M and T, since agents possess a certain ability to solve problems and, as the debate progresses, the answers from different Agents tend to become consistent, we can define the probability of obtaining a answer different with other agents is p. Therefore, the token cost will only increase to be comparable to that of Group Debate when different answers are obtained in each round, which occurs with a probability of only p^{MN}.

4 Experiments

4.1 Experimental Setup

Tasks and Metrics. To evaluate the effectiveness and efficiency of S^2-MAD in mathematical and logical reasoning tasks, we use total token cost and accuracy (ACC) as evaluation metrics across five representative tasks: (1) GSM8K (Cobbe et al., 2021): a dataset designed to assess the model's reasoning ability in complex mathematical problems. (2) MATH (Hendrycks et al., 2021): a dataset covers various branches of mathematics to evaluate the capacity to generate problem-solving logic and reasoning processes. (3) MMLU (Hendrycks et al., 2020): a dataset that aimed at evaluating the model's overall performance across diverse tasks. (4) GPQA (Rein et al., 2023): a multiple-choice question dataset, GPQA includes 448 questions across various disciplines. (5) Arithmetic (Brown et al., 2020): evaluates the model's performance in basic arithmetic operations to test its fundamental mathematical reasoning abilities.

Baselines. We compare our S^2-MAD with the following baselines: (1) Chain-of-Thought (CoT) (Wei et al., 2022); (2) Self-Consistency with Chain-of-Thought (CoT-SC) (Wang et al., 2022); (3) Multi-Agent Debate (MAD) (Liang et al., 2023); (4) Sparse MAD (S-MAD) (Li et al., 2024); (5) GroupDebate (GD) (Liu et al., 2024).Experiments are conducted with different numbers of agents, rounds, and group strategies. For example, (5,4) represents using 5 agents and 4 rounds, while CoT-SC(40) indicates CoT-SC with 40 reasoning paths.

Implementation Details. We set the number of intra-group discussion rounds to 2 and use a forgetting mechanism to retain the outputs from the previous round only. At the end of each intra-group discussion phase, we filter and summarize the results from the same groups. Our experiments use GPT-3.5-turbo-0301, GPT-4-0613 and Llama-3.1-8B-Instruct as agents, evaluating all baselines and our S^2-MAD in a zero-shot setting. Since the accuracy rate of the Arithmetic dataset reached 100% in a single GPT-4, no further comparison was conducted. Details about the prompts are explained in the Appendix C.

4.2 Main Result

In this section, we conducted a detailed comparison of our method with multi-agent debate methods (including MAD, S-MAD, GD) and single-agent methods (including CoT, CoT-SC). The main observations are as follows: firstly, we conduct a comparison between S^2-MAD with MAD. The results presented in Table 1 shows that S^2-MAD consistently reduces total token costs across different models while maintaining comparable accuracy, compared to MAD, it achieves a reduction of 94.5%, 84.2%, 92.4%, 83.6% and 88.7% on the five datasets, respectively. The variation in these percentages is due to the difficulty of the questions, which affects the model's performance. Furthermore, compared to S-MAD and GD, our approach achieves up to 90.2% and 87.0% less token cost, respectively. This demonstrates that there is a significant amount of redundancy in the information exchange during multi-agent debate, leading to the inefficiency of token costs throughout the debate process.

We also conducted a comparison with the single-agent method CoT, achieving a significant improvement in accuracy across five datasets, especially achieving up to 19.3% and 12.6% on GPQA and MMLU dataset. Furthermore, in comparison with the Cot-SC method, we successfully reached or exceeded Cot-SC's performance on certain datasets, such as GPQA and Arithmetic, while using relatively fewer token costs.

Methods	GSM8K		MATH		MMLU		GPQA		Arithmetic	
	ACC(%)↑	Tokens(k)↓	ACC(%)↑	Tokens(k)↓	ACC(%)↑	Tokens(k)↓	ACC(%)↑	Tokens(k)↓	ACC(%)↑	Tokens(k)↓
GPT-3.5-turbo-0125										
CoT	78.8±0.04	0.25±0.03	35.2±0.02	0.37±0.01	72.9±0.02	0.24±0.00	31.2±0.03	2.02±0.02	82.2±0.04	0.16±0.02
CoT-SC(40)	**85.6**±0.01	10.0±0.02	**48.2**±0.01	14.6±0.15	**78.4**±0.01	9.64±0.03	32.0±0.01	80.5±0.27	95.0±0.01	6.25±0.13
MAD(5,4)	83.6±0.01	20.4±0.12	40.5±0.00	23.4±0.24	74.1±0.03	26.9±0.11	41.4±0.03	64.3±3.06	96.2±0.02	20.3±0.32
S-MAD(5,4)	85.6±0.02	17.9±0.22	40.3±0.01	19.0±0.17	74.3±0.01	22.2±0.23	45.0±0.00	57.3±0.77	96.8±0.01	9.83±0.06
GD(5,4)	85.4±0.03	15.5±0.08	40.7±0.01	18.7±0.13	76.0±0.02	16.1±0.06	**45.0**±0.03	50.5±0.67	99.3±0.00	13.7±0.64
S^2-MAD(5,4)	84.8±0.02	4.53±0.31	40.3±0.00	11.4±0.29	76.8±0.02	4.78±0.57	43.8±0.01	23.6±7.05	**99.6**±0.00	2.29±0.07
GPT-4-0613										
CoT	92.8±0.01	0.38±0.00	73.0±0.01	0.71±0.00	83.7±0.01	0.39±0.00	47.2±0.02	2.23±0.10	-	-
CoT-SC(40)	**94.3**±0.00	15.2±0.03	**81.0**±0.01	27.9±0.00	88.4±0.01	15.4±0.02	53.0±0.00	90.0±0.30	-	-
MAD(5,4)	93.3±0.01	50.4±0.19	78.7±0.00	70.9±0.20	90.8±0.00	61.7±0.09	59.7±0.01	109.6±5.22	-	-
S-MAD(5,4)	93.0±0.02	24.4±0.04	79.3±0.00	59.5±0.53	**91.5**±0.00	48.2±0.24	60.7±0.02	97.8±0.42	-	-
GD(5,4)	94.3±0.00	21.4±0.08	76.7±0.01	36.8±0.28	87.8±0.01	25.3±0.11	**62.7**±0.02	64.6±1.25	-	-
S^2-MAD(5,4)	94.2±0.00	2.78±0.16	77.3±0.01	11.2±0.79	88.1±0.01	4.71±0.35	60.8±0.04	27.1±9.5	-	-
Llama-3.1-8B-Instruct										
CoT	83.2±0.01	0.32±0.00	32.0±0.03	0.53±0.00	61.2±0.04	0.43±0.00	19.7±0.02	2.35±0.01	74.0±0.01	0.19±0.00
CoT-SC(40)	**89.0**±0.01	12.6±0.00	43.0±0.01	21.1±0.02	74.1±0.02	17.2±0.04	33.5±0.02	93.9±0.01	83.0±0.01	7.63±0.00
MAD(5,4)	86.7±0.02	31.4±0.27	**46.0**±0.01	80.5±0.20	73.4±0.02	54.7±0.58	37.0±0.01	117.5±4.23	**91.0**±0.02	76.1±0.12
S-MAD(5,4)	87.3±0.02	26.0±0.36	45.0±0.00	66.3±0.43	**74.5**±0.00	43.3±0.14	**40.0**±0.01	102.2±2.53	89.5±0.01	62.3±0.53
GD(5,4)	86.5±0.00	17.0±0.07	44.0±0.01	39.9±0.28	73.5±0.01	39.5±0.11	37.0±0.02	71.6±2.25	89.3±0.03	33.4±0.02
S^2-MAD(5,4)	85.7±0.02	5.39±0.16	44.0±0.05	21.9±0.17	73.8±0.04	10.6±1.74	39.0±0.04	19.3±1.5	90.0±0.03	13.9±0.29

Table 1: **Comparison of Token Cost and Accuracy Between S^2-MAD and Other Methods.** The results of highest accuracy are **bold** and the results of both highest accuracy and lowest token cost except from CoT are underlined. The dash (-) indicates that the model achieved a correctness rate of 1 for all methods on this dataset.

4.3 In-Depth Analysis

Group Strategy. To assess the impact of diverse grouping strategies on performance and token cost, we conducted experiments involving 8 agents across 3 rounds using the GSM8K dataset. As shown in the Table 2, increasing the number of groups can reduce the total token cost as the quantity of information exchange is limited by communication constraints. However, when the number of agents within a group increases, agents can more effectively receive diverse information, achieving higher accuracy. Our findings suggest that there is a clear trade-off between optimizing token consumption and maintaining high accuracy, highlighting the importance of selecting an appropriate grouping strategy.

Agent, Round and Token Cost Scaling. To assess the impact of the number of rounds and agents on the accuracy and token cost of different methods, we analyze the trends in accuracy and token cost cross different combinations of rounds and agents. As shown in the Figure 3, with an increase in the number of agents and rounds, there is a noticeable enhancement in the overall performance of various methods; however, this also leads to a significant increase in token cost. Our approach maintains a certain level of performance while exhibiting a gradual increase in token cost as agents and rounds increase, achieving the lowest token cost across different setting, as shown in Figure 4. This indicates that there is a significant amount of redundant and repetitive information exchange during debates, resulting in higher token costs and less effective agent debates.

4.4 Ablation Study

To investigate the impact of different modules and strategies on performance, we conducted the ablation experiment on the GSM8K dataset using GPT-3.5-turbo, are shown in Table 3. Our S^2-MAD achieved an accuracy of 85.6% with a token cost of 4.73k, demonstrating a significant 72.7% reduction in token usage compares to MAD. We also have a try to further leverage sparsity, however, using the addition of sparse communication slightly decreased accuracy to 84.7%, while retaining a similar token cost. Without the early stopping strategy led to a slight accuracy drop to 84.4% but maintained a comparable token cost of 4.99k. In contrast, removing the jump strategy resulted in a more substantial decline in accuracy to 80.8% and an in-

Method	ACC (%)	Token (k)	Cost Saving
MAD(8,3)	86.7±0.02	28.5±0.08	-
S-MAD(8,3)	86.5±0.01	18.7±0.04	-34.4%
GD			
2+6	86.7±0.00	20.3±0.08	-28.8%
4+4	87.3±0.01	19.1±0.00	-33.0%
2+3+3	87.3±0.02	18.0±0.05	-36.8%
2+4+4	87.7±0.00	18.3±0.03	-35.7%
2+2+2+2	**87.8**±0.00	17.4±0.04	-38.9%
S^2-MAD			
2+6	84±0.00	8.01±0.13	-71.9%
4+4	84.6±0.00	7.28±0.19	-74.5%
2+3+3	85.1±0.00	6.97±0.14	-75.5%
2+4+4	84.5±0.00	7.02±0.20	-75.4%
2+2+2+2	83.4±0.02	**6.78**±0.12	-76.2%

Table 2: **Comparison of different group strategies with GD and S^2-MAD on GSM8K datasets.** The notation 2+6 signifies two distinct groups containing 2 and 6 agents respectively.The results of highest accuracy are **bold** and the results of both highest accuracy and lowest token cost are underlined.

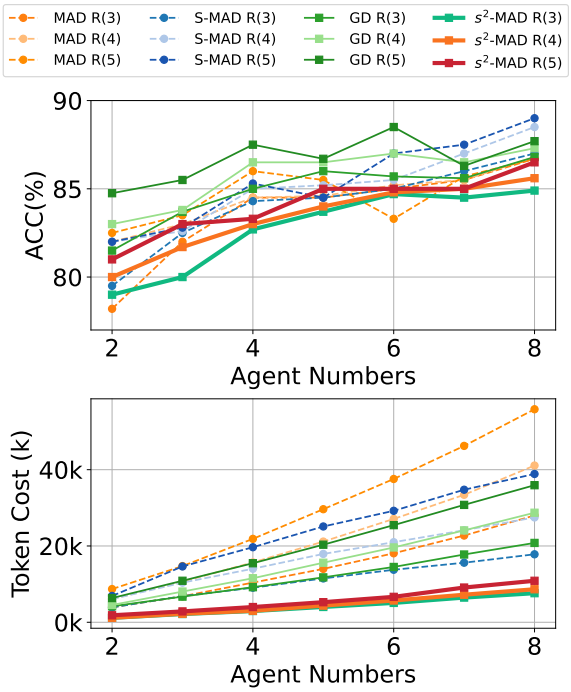

Figure 3: **Scaling study of Agents and Rounds.**

Method	ACC (%)	Token (k)
S^2-MAD	85.6±0.00	4.73 (-72.7%)
+sparse	84.7±0.00	4.71 (-73.8%)
-Early Stop	84.4±0.00	4.99 (-72.3%)
-Jump	80.8±0.02	9.45 (-47.5%)
-Filter	87.6±0.01	13.4 (-25.6%)
MAD	85.4±0.02	18

Table 3: **Comparison of accuracy and cost saving againts MAD on GSM8K dataset.** All experiments were conducted suing GPT-3.5-turbo.

crease in token usage to 9.45k, We hypothesize this may be due to insufficient information diversity, causing redundant checks that affect response accuracy. Finally, although removing the filtering module can increase accuracy to 87.6%, it also leads to an increase in token cost of 13.4k. Although our method has not yet achieved optimal performance in terms of accuracy and token cost, it still shows a slight improvement over the MAD method while significantly reducing token usage, highlighting the efficiency of our proposed method in balancing accuracy and token savings.

5 Related Work

5.1 LLM Reasoning

Many studies have explored ways to improve the logical reasoning abilities of LLMs. CoT (Wei et al., 2022) mimics human thought processes by breaking complex tasks into sequential steps. Many CoT variants (Zelikman et al., 2022; Wang et al., 2022; Shum et al., 2023) extend this framework by generating multiple reasoning chains and selecting the optimal one based on specific criteria. Building on this, Tree-of-Thoughts (ToT) (Yao et al., 2024) structures the reasoning process into a tree-like path, where each step serves as a decision point, enabling the evaluation of multiple reasoning paths and self-assessment. Similarly, Skeleton-of-Thought (Ning et al., 2023) accelerates answer generation by first creating a skeletal framework and then completing the content in parallel for each point. Table-of-Thoughts (Jin and Lu, 2023) improves reasoning accuracy through structured modeling of the reasoning process. While these CoT-based methods follow structured reasoning paths, more complex reasoning structures have been proposed. For instance, Graph-of-Thoughts (Besta et al., 2024) models reasoning as a flexible graph, allowing for non-linear task solving beyond the limitations of chains or trees. Methods such as Least-to-Most (Zhou et al., 2022) and Lambada (Kazemi et al., 2022) take a problem decomposition approach, breaking tasks into subproblems and solv-

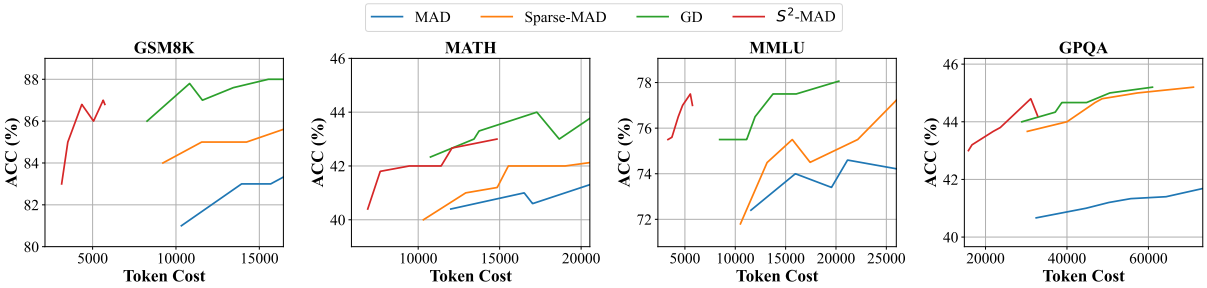

Figure 4: **Scaling Study of Token Cost.**

ing them step-by-step, where each sub-answer informs the next step. Additionally, frameworks like LReasoner (Wang et al., 2021) introduce mechanisms that enhance reasoning by extracting logical structures embedded in the problem. Logic-LM (Pan et al., 2023) combines symbolic solvers to convert natural language into symbolic formulas and introduces a self-refinement module to correct errors during the reasoning process.

5.2 Multi-agent Debate

MAD is a promising approach to enhance the reasoning capabilities of LLMs by facilitating discussions among multiple agents who collaboratively refine and update generated answers. (Liang et al., 2023) presents a MAD framework where multiple agents engage in "tit for tat" argumentation, managed by a judge, to stimulate divergent thinking in LLMs. Building on this foundation, (Xiong et al., 2023) introduce the FORD framework, which organizes a three-stage debate aligned with real-world scenarios, comprising fair debate, mismatched debate, and round-table debate formats. (Xu et al., 2023) present a framework that mirrors the academic peer review process, allowing models to autonomously develop solutions, review each other's work, and revise their answers based on feedback. ChatEval (Chan et al., 2023), another MAD framework, employs diverse communication strategies and varied role prompts to foster human-like interactions and evaluations in natural language dialogue. Moreover, (Wang et al., 2023) address cognitive constraints in multi-agent debates by integrating prior knowledge retrieval and a self-selection module, enhancing reasoning capabilities and overall performance. Further exploring collaboration, (Fu et al., 2023) analyze the autonomous enhancement of negotiation strategies among LLMs through role-playing and iterative AI feedback within a structured negotiation game, highlighting the trade-offs between deal quality and risk management. However, as the number of agents and debate rounds increases, token costs can rise significantly. To mitigate this, (Du et al., 2023) suggests summarizing agent outputs at the end of each round for subsequent inputs, and (Sun et al., 2023) introduces a "forgetfulness" mechanism to retain only the previous round's output. The MAD-Sparse approach (Li et al., 2024) utilizes a sparse communication strategy, limiting information exchange to adjacent agents. Additionally, GroupDebate (Liu et al., 2024) promotes a grouping strategy, allowing agents to debate internally while sharing interim results. However, these methods do not enable agents to critically assess the redundancy of incoming information, limiting overall efficiency.

6 Conclusion

In this work, we identified the issue of redundant viewpoints among agents in Multi-agent Debate (MAD). To address this, we proposed **S**elective **S**parse **M**ulti-**A**gent **D**ebate (S^2-MAD), a novel strategy designed to reduce token cost by selectively incorporating non-redundant viewpoints from different agents, thereby significantly improving the efficiency of information exchange and debate. Our theoretical analyses verify the effectiveness of S^2-MAD, and extensive experiments conducted on five benchmark datasets demonstrate that S^2-MAD can significantly reduce token cost in MAD while maintaining competitive performance. For future work, we aim to refine S^2-MAD by further optimizing the identification and condensation of non-redundant viewpoints between agents, with the goal of further reducing token cost and enhancing efficiency. Additionally, exploring methods to increase the diversity of thought among agents will be key to improving the overall accuracy of S^2-MAD.

7 Limitation

Despite the significant reduction in token cost achieved by S^2-MAD, our method has several limitations. First, the reduction in token cost exhibits variability depending on the consistency of agent responses. When agents' answers differ significantly, the efficiency gains are limited, whereas more consistent responses yield a greater reduction in token cost. This variability introduces an element of unpredictability to the system's overall efficiency. Second, the judge module in S^2-MAD is sometimes unable to filter out redundant viewpoints. The module relies on keyword extraction using regular expressions to determine whether agents' outputs convey the same idea. However, when agents express similar views with different wording or synonyms, the judge module may fail to detect these similarities, resulting in redundant exchanges of information. This can undermine the potential gains in efficiency and contribute to token cost redundancy. Therefore, there remains room for improvement in optimizing the token cost of S^2-MAD.

References

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

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*.

Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. 2023. Improving language model negotiation with self-play and in-context learning from ai feedback. *arXiv preprint arXiv:2305.10142*.

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

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

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

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

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

Mehran Kazemi, Najoung Kim, Deepti Bhatia, Xin Xu, and Deepak Ramachandran. 2022. Lambada: Backward chaining for automated reasoning in natural language. *arXiv preprint arXiv:2212.13894*.

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

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.

Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. 2023. Evaluating the logical reasoning ability of chatgpt and gpt-4. *arXiv preprint arXiv:2304.03439*.

Tongxuan Liu, Xingyu Wang, Weizhe Huang, Wenjiang Xu, Yuting Zeng, Lei Jiang, Hailong Yang, and Jing Li. 2024. Groupdebate: Enhancing the efficiency of multi-agent debate using group discussion. *arXiv preprint arXiv:2409.14051*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Huazhong Yang, and Yu Wang. 2023. Skeleton-of-thought: Large language models can do parallel decoding. *arXiv preprint arXiv:2307.15337*.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. *arXiv preprint arXiv:2305.12295*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof qa benchmark. *Preprint*, arXiv:2311.12022.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

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

Kaya Stechly, Matthew Marquez, and Subbarao Kambhampati. 2023. Gpt-4 doesn't know it's wrong: An analysis of iterative prompting for reasoning problems. *arXiv preprint arXiv:2310.12397*.

Qiushi Sun, Zhangyue Yin, Xiang Li, Zhiyong Wu, Xipeng Qiu, and Lingpeng Kong. 2023. Corex: Pushing the boundaries of complex reasoning through multi-model collaboration. *arXiv preprint arXiv:2310.00280*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Karthik Valmeekam, Matthew Marquez, and Subbarao Kambhampati. 2023. Can large language models really improve by self-critiquing their own plans? *arXiv preprint arXiv:2310.08118*.

Haotian Wang, Xiyuan Du, Weijiang Yu, Qianglong Chen, Kun Zhu, Zheng Chu, Lian Yan, and Yi Guan. 2023. Apollo's oracle: Retrieval-augmented reasoning in multi-agent debates. *arXiv preprint arXiv:2312.04854*.

Siyuan Wang, Wanjun Zhong, Duyu Tang, Zhongyu Wei, Zhihao Fan, Daxin Jiang, Ming Zhou, and Nan Duan. 2021. Logic-driven context extension and data augmentation for logical reasoning of text. *arXiv preprint arXiv:2105.03659*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2022. Generating sequences by learning to self-correct. *arXiv preprint arXiv:2211.00053*.

Kai Xiong, Xiao Ding, Yixin Cao, Ting Liu, and Bing Qin. 2023. Examining inter-consistency of large language models collaboration: An in-depth analysis via debate. *arXiv preprint arXiv:2305.11595*.

Jin Xu, Xiaojiang Liu, Jianhao Yan, Deng Cai, Huayang Li, and Jian Li. 2022. Learning to break the loop: Analyzing and mitigating repetitions for neural text generation. *Advances in Neural Information Processing Systems*, 35:3082–3095.

Zhenran Xu, Senbao Shi, Baotian Hu, Jindi Yu, Dongfang Li, Min Zhang, and Yuxiang Wu. 2023. Towards reasoning in large language models via multi-agent peer review collaboration. *arXiv preprint arXiv:2311.08152*.

Jianhao Yan, Jin Xu, Chiyu Song, Chenming Wu, Yafu Li, and Yue Zhang. 2023. Understanding in-context learning from repetitions. *arXiv preprint arXiv:2310.00297*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A Selective Sparse MAD Algorithm

The detailed S^2-MAD Algorithm is as follows:

Algorithm 1 S^2-MAD Methods

Require: Number of groups N, number of agents M, question Q, total rounds T, intra-group debate round R, total stages S, redundancy filter F, Opinion Judger J, answer extracter $VOTE$

Ensure: $Answer$

$A \leftarrow [A_1, A_2, \ldots, A_M]$
▷ Initialize and shuffle the agents randomly
$G \leftarrow [G_1, G_2, \ldots, G_N]$ ▷ Initialize each group
$H \leftarrow [H_1, H_2, \ldots, H_M]$
▷ Initialize each agent with empty memory
$Sum \leftarrow [Sum_1, Sum_2, \ldots, Sum_N]$
for $i = 1$ to M **do**
  $H_i \leftarrow [Q]$
end for
for $s = 1$ to S **do**
  for $j = 1$ to N **do**
    for $t = (s-1)R + 1$ to $\min(sR, T)$ **do**
      for $A_i \in G_j$ **do**
        if $s = 1$ **and** $t = 1$ **then**
          $h_i \leftarrow A_i(H_i)$
          $H_i \leftarrow H_i + h_i$
          $H_i \leftarrow H_i + BUF$
        else
          $buf \leftarrow [\ \ ]$
          if $s \neq 1$ **and** $t = (s-1)R+1$ **then**
            for $S_i \in Sum$ **do**
              if $J(H_i[-2], S_i)$ **then**
                $buf \leftarrow buf + S_i$
              end if
            end for
          else
            for $A_{i'} \in G_j$ **and** $A_{i'} \neq A_i$ **do**
              if $J(H_i[-2], H_{i'}[-2])$ **then**
                $buf \leftarrow buf + Replay_{i'}$
              end if
            end for
          end if
          if $len(buf) \neq 0$ **then**
            $H_i[-1] \leftarrow buf$
            $h_i \leftarrow A_i(H_i)$
            $H_i[-2] \leftarrow h_i$
          end if
        end if
      end for
    end for
    if $s \neq S$ **then**
      $summary \leftarrow [\,]$
      for $A_i \in G_j$ **do**
        $sum \leftarrow sum + H_i[-2]$
      end for
      $Sum_j \leftarrow LLM(sum)$
    end if
  end for
  $Sum \leftarrow F(Sum)$
  if $len(Sum) = 1$ **then** break ▷ End debate if only one summary
  end if
end for
$Answer \leftarrow VOTE(H)$
return $Answer$

B Token Cost Analysis

In this appendix section, we aim to provide a theoretical analysis of the token cost for S^2-MAD. As LLMs' outputs typically are not too long and we can actually control the token length of LLMs' outputs in prompts to some extent, we assume

that the upper bound on the number of tokens output by each agent participating in debate is $Output_{max}$ and the upper bound on the number of tokens in the generated summary is $Summary_{max}$. We define C as the maximum of $Output_{max}$ and $Summary_{max}$, P represents the upper bound of the average probability of each Agent participating in the debate globally.

As mentioned in Section 3, our S^2-MAD includes three types of processes and thus the total token cost $Token^{GD}$ can be further dividied into:

$$
\begin{aligned}
Token = & \underbrace{Token_1^1}_{\text{initial thinking}} + \\
& \underbrace{\sum_{s=2}^{S}(Token_{s-1}^{summary} + Token_s^{(s-1)R+1})}_{\text{inter-group discussion}} \\
& + \underbrace{\sum_{s=1}^{S}\sum_{t=(s-1)R+2}^{min(sR,T)} Token_s^t}_{\text{intra-group discussion}}
\end{aligned} \tag{5}
$$

Specifically, for initial thinking, the token cost of each agent includes the initial question prompt and its own output. For intra-group debate, the token cost of each agent includes the unique responses from other agents within the same group that differ from its own in the previous round and its output. For inter-group debate, the token cost includes the summary generation cost, which comprises the unique responses from all groups and the output summary, as well the token cost of each agent which comprises as its output and summary from from other groups that differ from its own. The detailed computation process of the token cost in S^2-MAD can be found in Algorithm 2.

Following Alogorithm 2 and Eq. B, we have:

$$
\begin{aligned}
Token = & MQ + \sum_{i=1}^{M} O_i^1 \\
& + \sum_{s=2}^{S}[\sum_{j=1}^{N}(\sum_{i\in G_j} O_i^{(s-1)R} + Sum_j^{s-1}) \\
& + \sum_{i=1}^{M} P_i^{(s-1)R}(Q + O_i^{(s-1)R+1} \\
& \quad + \frac{D_i^t}{N}\sum_{j=1}^{N} Sum_j^{s-1} + Output_i^{(s-1)R+1})] \\
& + \sum_{s=1}^{S}\sum_{t=(s-1)R+2}^{min(sR,T)}\sum_{j=1}^{N}\sum_{i\in G_j} P_i^t(Q + O_i^t \\
& \qquad + \frac{ND_i^t}{M}\sum_{i'\in G_j} O_{i'}^{t-1}) \\
& \leq MTQ + P_{max} \times \{ \\
& \quad [3MS - 2M + (T-S)(K+1)M] \times O_{max} \\
& \quad + (S-1)(M+1)N \times Sum_{max}\} \\
& \leq MTQ + P_{max} \times \{\frac{2M^2T}{N} \times O_{max} \\
& \quad + 2MSN \times Sum_{max}\} \\
& = \mathcal{O}\left(MTQ + (\frac{M^2T}{N} + MSN)CP\right)
\end{aligned} \tag{6}
$$

When we set $N \rightarrow \mathcal{O}\left(\sqrt{\frac{MT}{S}}\right)$, we can theoretically obtain $Token \rightarrow \mathcal{O}\left(MTQ + \sqrt{M^3TS}CP\right)$. Furthermore, If we consider setting S to a very small positive integer and the average probability of their participation decreases as the capability of individual agents improves, then $Token$ can approach $\mathcal{O}\left(MTQ + \sqrt{M^3T}CP\right)$. This complexity is significantly lower than that of MAD.

C Prompts

In this section, we present some examples of prompts. Table 4 displays the input prompts used in our S^2-MAD across different datasets, which encompass five different types. Table 5 outlines the prompts regarding output Format Requirements in our S^2-MAD.

Type	Task	Prompt
System	All	*Welcome to the debate! You are a seasoned debater with expertise in succinctly and persuasively expressing your viewpoints. You will be assigned to debate groups, where you will engage in discussions with fellow participants. The outcomes of each group's deliberations will be shared among all members. It is crucial For you to leverage this inFormation effectively in order to critically analyze the question at hand and ultimately arrive at the correct answer. Best of luck!*
Starting	Arithmetic	*What is the result of {}+{}*{}+{}-{}*{}? < Output Format >.*
	GSM8K	*Can you solve the following math problem? <Problem> Explain your reasoning. < Output Format >.*
	MMLU	*Can you answer the following question? <Problem>: A) , B) , C) , D) Explain your answer, <Output Format>.*
	MATH	*Can you solve the following math problem? <Problem> Explain your reasoning as concise as possible. <Output Format> .*
	GPQA	*Can you answer the following question? <Problem>: A) , B) , C) , D) Explain your answer, <Output Format>.*
Intra-group Debate	All	*These are the recent unique opinions from other agents that differ with yours: <other agent responses> Using the opinions carefully as additional advice, can you provide an updated answer? Examine your solution and that other agents step by step. <Output Format> .*
Summary	All	*These are the recent/updated and unique opinions from all agents: <all agent responses> Summarize these opinions carefully and completly in no more than 80 words. Aggregate and put your final answers in parentheses at the end of your response.*
Inter-group Debate	All	*These are the recent unique opinions from all groups: one group responses: <group summary>. Using the reasoning from all groups as additional advice, can you give an updated answer? Examine your solution and that all groups step by step. <Output Format>.*

Table 4: **Prompts in Each Stage.** List of prompts used in each task.

Dataset	Output Format Requirements
Arithmetic	*Make sure to State your answer at the end of the response.*
GSM8K	*Your final answer should be a single numerical number, in the Form \boxed{{answer}}, at the end of your response.*
MMLU	*Put your final choice in parentheses at the end of your response.*
MATH	*Put your final answer in the Form \boxed{{answer}}, at the end of your response.*
GPQA	*Put your final answer in the Form \The correct answer is (insert answer here)*

Table 5: **Output Format Requirements in Each Dataset.**

Algorithm 2 Tokens Cost in S^2-MAD Methods

Require: Number of groups N, number of agents M, question length Q, total rounds T, group debate round R, total stages S, summary of each group at the end of each stage $Summary = \{Summary_j^s | j = 1, 2, \ldots, N, s = 1, 2, \ldots, S\}$, output length of each agent $A_i (i = 1, 2, \ldots, M)$ in each round $t (t = 1, 2, \ldots, T)$ $Output_i^t$, each group agents set $G = \{G_j | j = 1, 2, \ldots, N\}$, probability of participating in the debate of each agent $P = \{P_i^t | i = 1, 2, \ldots, M, t = 1, 2, \ldots, T\}$.

Ensure: Total token cost $Token$

$Token_1^1 \leftarrow M \times Q + \sum_{i=1}^{M} O_i^1$

▷ First round

for $t = 2$ to R **do**

$Token_1^t \leftarrow \sum_{j=1}^{N} \sum_{i \in G_j} (Q + O_i^t + \sum_{i' \in G_j} O_{i'}^{t-1})$

▷ Subsequent rounds of the first stage

end for

for $s = 2$ to S **do**

$Token_{s-1}^{summary} \leftarrow \sum_{j=1}^{N} (\sum_{i \in G_j} O_i^{(s-1)R} + Sum_j^{s-1})$

▷ Summary at the end of stage $s - 1$

$Token_s^{(s-1)R+1} \leftarrow \sum_{i=1}^{M} P_i^{(s-1)R+1} (Q + O_i^{(s-1)R} + \sum_{j=1}^{N} Sum_j^{s-1} + O_i^{(s-1)R+1})$

▷ First round of the stage s

for $t = (s-1)R + 2$ to $\min(sR, T)$ **do**

$Token_s^t \leftarrow \sum_{j=1}^{N} \sum_{i \in G_j} P_i^t (Q + O_i^t + \sum_{i' \in G_j} O_{i'}^{t-1})$

▷ Subsequent rounds of the stage s

end for

end for

$Token^{GD} \leftarrow \sum_{t=1}^{R} Token_1^t + \sum_{s=2}^{S} (Token_{s-1}^{summary} + \sum_{t=(s-1)R+1}^{\min(sR,T)} Token_s^t)$

▷ Total token cost in debate

return $Token$