CortexDebate: Debating Sparsely and Equally for Multi-Agent Debate

Anonymous ACL submission

Abstract

Nowadays, single Large Language Model (LLM) struggles with critical issues such as hallucination and inadequate reasoning abilities. To mitigate these issues, Multi-Agent Debate (MAD) has emerged as an effective strategy, where LLM agents engage in in-depth debates with others on tasks. However, existing MAD methods face two major issues: (a) *too lengthy input contexts*, which causes LLM agents to get lost in plenty of input information and experiences performance drop; and (b) *the overconfidence dilemma*, where self-assured LLM agents dominate the debate, leading to low debating effectiveness. To address these limitations, we propose a novel MAD method called "CortexDebate". Inspired by the human brain's tendency to establish a sparse and dynamically optimized network among cortical areas governed by white matter, CortexDebate constructs a sparse debating graph among LLM agents, where each LLM agent only debates with the ones that are helpful to it. To optimize the graph, we propose a module named McKinsey-based Debate Matter (MDM), which acts as an artificial analog to white matter. By integrating the McKinsey Trust Formula, a well-established measure of trustworthiness from sociology, MDM enables credible evaluations that guide graph optimization. The effectiveness of our CortexDebate has been well demonstrated by extensive experimental results across eight datasets from four task types.

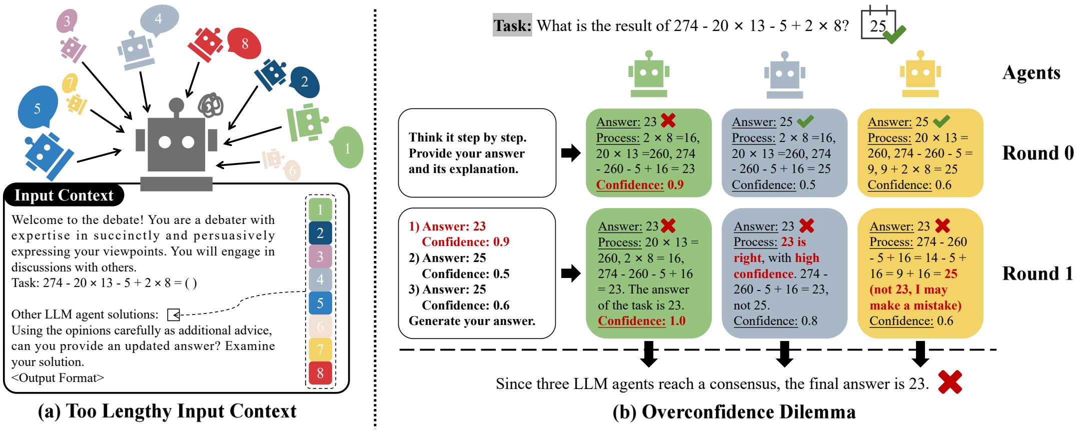
1 Introduction

Recently, inspired by human cooperation, many multi-agent interaction methods (Wan et al., 2024; Xu et al., 2023a; Tu et al., 2023; Hu et al., 2024) have been proposed to further improve the reasoning results of LLMs. These methods aim to address critical issues faced by single LLM agent, such as hallucination and poor reasoning ability. Among these methods, Multi-Agent Debate (MAD) (Zhang et al., 2024a; Du et al., 2023) has emerged as one of the most promising strategies, as it can effectively improve the performance of LLM agents through the debating process among them.

Although previous MAD methods have achieved promising results, they still suffer from two major shortcomings. As shown in Figure 1, firstly, in these methods, each LLM agent is often required to debate with all other LLM agents, which causes its input context to expand significantly as the number of agents and debating rounds increase. Consequently, since single LLM agent usually struggles to handle lengthy input contexts (Liu et al., 2024a), it may get lost in the vast amount of input information, leading to a significant performance drop. Secondly, prior MAD methods determine the debating influence of each LLM agent simply according to its own confidence, which may lead to the overconfident LLM agents gradually dominating the entire debating process. As a result, the potential useful information provided by other "weak" LLM agents may be ignored. Such unequal debate is harmful to debating effectiveness, as also confirmed by (Xiong et al., 2023; Xu et al., 2023b).

Therefore, inspired by the human cognition theory (Thiebaut de Schotten and Forkel, 2022), this paper proposes a new MAD approach named CortexDebate, which mimics the working mode of the human brain cortex. As revealed by (Thiebaut de Schotten and Forkel, 2022), given a problem, the human brain tends to establish a dynamic and sparse network among different cortical areas, and this network is gradually optimized by a specialized module named white matter. During the optimization process, the white matter focuses more on the influence between paired areas rather than the performance of a single cortical area.

By treating LLM agents as cortical areas in human brain, our CortexDebate establishes a sparse and directed debating graph, where the nodes represent LLM agents and the edges carry information transmission. Each directed graph edge is assigned



3
4
8
7
2
5
1
6
Input Context
Welcome to the debate! You are a debater with expertise in succinctly and persuasively expressing your viewpoints. You will engage in discussions with others.
Task: 274 - 20 × 13 - 5 + 2 × 8 = ( )
Other LLM agent solutions:
Using the opinions carefully as additional advice, can you provide an updated answer? Examine your solution.
<Output Format>
1
2
3
4
5
6
7
8
Task: What is the result of 274 - 20 × 13 - 5 + 2 × 8?
25
Agents
Think it step by step. Provide your answer and its explanation.
Answer: 23 Process: 2 × 8 =16, 20 × 13 =260, 274 - 260 - 5 + 16 = 23 Confidence: 0.9
Answer: 25 Process: 2 × 8 =16, 20 × 13 =260, 274 - 260 - 5 + 16 = 25 Confidence: 0.5
Answer: 25 Process: 20 × 13 = 260, 274 - 260 - 5 = 9, 9 + 2 × 8 = 25 Confidence: 0.6
Round 0
1) Answer: 23 Confidence: 0.9
2) Answer: 25 Confidence: 0.5
3) Answer: 25 Confidence: 0.6
Generate your answer.
Answer: 23 Process: 20 × 13 = 260, 2 × 8 = 16, 274 - 260 - 5 + 16 = 23. The answer of the task is 23. Confidence: 1.0
Answer: 23 Process: 23 is right, with high confidence. 274 - 260 - 5 + 16 = 23, not 25. Confidence: 0.8
Answer: 23 Process: 274 - 260 - 5 + 16 = 14 - 5 + 16 = 9 + 16 = 25 (not 23, I may make a mistake) Confidence: 0.6
Round 1
Since three LLM agents reach a consensus, the final answer is 23.

(a) Too Lengthy Input Context

(b) Overconfidence Dilemma

Figure 1: Shortcomings of the existing MAD methods: (a) Debating with all others causes lengthy contexts input to LLM agents. They may get lost in the vast amount of input information and perform unsatisfactorily; (b) Determining the debating impact of LLM agents simply based on their self-confidence may lead to the overconfident ones dominating the debate. This situation is harmful to the debating performance.

a weight that reflects how much the performance of the tail LLM agent is expected to be improved by debating with the head LLM agent. Therefore, each tail LLM agent will not debate with the head LLM agents which do not help improve its performance. It means that the edges with small weights in the debating graph will be removed, resulting in a sparse graph. As a result, the length of context input to such tail LLM agent will also be reduced. To optimize the edge weights of the debating graph, akin to the white matter dynamically governing the optimization of sparse graph among different cortical areas in human brain, our CortexDebate introduces a module named McKinsey-based Debate Matter (MDM) that serves as the artificial white matter. To alleviate the overconfidence dilemma present in prior works, MDM considers both the performance of head LLM agent and the performance improvement expectation of tail LLM agent in deciding each edge weight. Specifically, MDM innovatively introduces McKinsey Trust Formula (Lamarre et al., 2012) to calculate edge weights, which has been widely used in sociology to evaluate the level of trustworthiness of a person through four aspects, including credibility, reliability, intimacy, and self-orientation. Among them, the first two evaluate individual abilities, while the last two evaluate the collaboration effectiveness with others. Therefore, this formula may suppress overconfident LLM agents, and also balance individual competence with teamwork ability of LLM agents in MAD.

The effectiveness of our CortexDebate has been well confirmed by the experiments on diverse tasks, including math, world knowledge question answering, reasoning, and long-context understanding. For instance, when compared with the state-of-the-art methods, in math task, CortexDebate increases *Result Accuracy* (RA) by up to 9.00% on GSM-IC dataset and 10.00% on MATH dataset, respectively. In reasoning task, CortexDebate increases RA by up to 9.00% on GPQA dataset and 12.33% on ARC-C dataset, respectively. Besides, apart from achieving high performance, our CortexDebate significantly reduces the length of context input to each LLM agent, with a maximum reduction of 70.79%.

The main contributions of this paper are summarized as follows:

1) We propose a new MAD method named CortexDebate, which can improve the performance of LLM agents by establishing a sparse and dynamic debating graph and reducing the burden of lengthy input context during the debate.

2) We propose a new module named MDM, which introduces McKinsey Trust Formula to evaluate both the confidence of each LLM agent and the usefulness to its debating component, thereby alleviating the overconfidence of LLM agents.

3) We conduct extensive experiments to show that our proposed CortexDebate outperforms representative baseline methods across multiple tasks such as math, world knowledge question answering, reasoning, and long-context understanding.

2 Related Work

In a MAD system, each LLM agent presents its viewpoint and scrutinizes the viewpoints of other LLM agents across multiple rounds of debate (Sun et al., 2024a). In summary, the existing MAD methods can be categorized as two types, namely *sequential debate* and *parallel debate*.

Sequential Debate. In these methods (Hu et al., 2025; Brown-Cohen et al., 2023; Michael et al., 2023; Wang et al., 2025; He et al., 2024), LLM agents generate their viewpoints in turn. Each LLM agent can only obtain the viewpoints of its preceding LLM agents. For example, Liang et al. (2023) require two LLMs to refute each other in turn. In addition to debaters, Guan et al. (2025) add extra roles, such as judge and critic. The judge speaks before debaters to explain the task, and the critic speaks last to summarize debates. However, in a sequential debate system, each LLM agent must wait for previous LLM agents to finish reasoning before it starts. This makes debating time increase linearly with the number of LLM agents, leading to low efficiency which is fatal to multi-agent systems.

Parallel Debate. In these methods (Pham et al., 2023; Yin et al., 2023; Chern et al., 2024; Khan et al., 2024; Liang et al., 2024; Li et al., 2024a; Hegazy, 2024; Zhang et al., 2024b), all LLM agents simultaneously generate their viewpoints based on the viewpoints of other LLM agents in the last debating round. For example, Chan et al. (2023) require LLM agents to critique all answers generated in the last debating round and update its answer in each debating round simultaneously. In addition to the answers generated in the last round, Sun et al. (2024b) also provide each LLM agent with task-related information retrieved from the web. Besides, some methods (Duan and Wang, 2024; Yoffe et al., 2024) try to adjust the debating influence of each LLM agent to improve the debating effectiveness. For example, Chen et al. (2023) require each LLM agent to generate the confidence score for its own answer, and then inputs the score to other LLM agents along with the answer.

Since sequential debate systems face the low-efficiency issue mentioned above, our proposed CortexDebate follows the parallel debate framework. Compared with existing parallel debating methods which require each LLM agent to debate with all others in each round, our CortexDebate dynamically decides the necessary debating agents by establishing a sparse debating graph among all involved LLM agents, so that the input context to each agent can be shortened. This is also in contrary to (Liu et al., 2024b; Li et al., 2024b) in which the debating opponents are fixed. Furthermore, different from prior methods which determine the debating impact of each LLM agent simply based on its own confidence, we introduce the McKinsey Trust Formula so that both the confidence of each LLM agent and the usefulness to its debating component can be evaluated.

3 Preliminaries

In this section, we provide the problem definition for our CortexDebate, and introduce the McKinsey Trust Formula which plays an important role in our proposed CortexDebate.

3.1 Problem Definition

Our CortexDebate establishes a directed debating graph among n LLM agents, $\mathcal{G} = (\mathcal{A}, \mathcal{E})$, where $\mathcal{A} = \{A_i\}_{i=1}^n$ is the vertex set representing participating LLM agents and $\mathcal{E} = \{E_{i\to j}\}_{i,j\in[1,2,\ldots,n]}$ is the directed edge set representing information transmission. Here, each directed edge $E_{i\to j}$ is assigned a weight $W_{i\to j}$ that indicates the expected improvement in the performance of agent A_j by debating with A_i. All the weights $\{W_{i\to j}\}$ are dynamically optimized during the debate process. Given a problem Q, the agents $\{A_i\}_{i=1}^n$ engage in D rounds of debate. In the d-th debate round, each LLM agent A_i scrutinizes the outputs of the LLM agents connected to it, and then generates its own output O_i^d along with a self-confidence score H_i^d. Afterwards, the final answer of this debate round, *i.e.*, F_d, is obtained by majority voting.

3.2 McKinsey Trust Formula

The McKinsey Trust Formula (Lamarre et al., 2012) is widely used in sociology to evaluate the level of trustworthiness of a person within a group. This formula can be expressed as:

$$T = \frac{C \times R \times I}{S}, \tag{1}$$

where C, R, I, and S denote credibility, reliability, intimacy, and self-orientation, respectively. Among them, credibility measures professional competence, reliability measures the stability of task performance, intimacy measures the relationship with the evaluated person, and self-orientation measures the self-orientation level of the evaluated person within a group.

In our MDM module, we adapt these four factors to the context of MAD. Specifically, for directed edge $E_{i\to j}$ connecting agent A_i to A_j, credibility evaluates the professional competence of A_i. Reliability is the average confidence score of A_i to its own answers in history debates, which represents

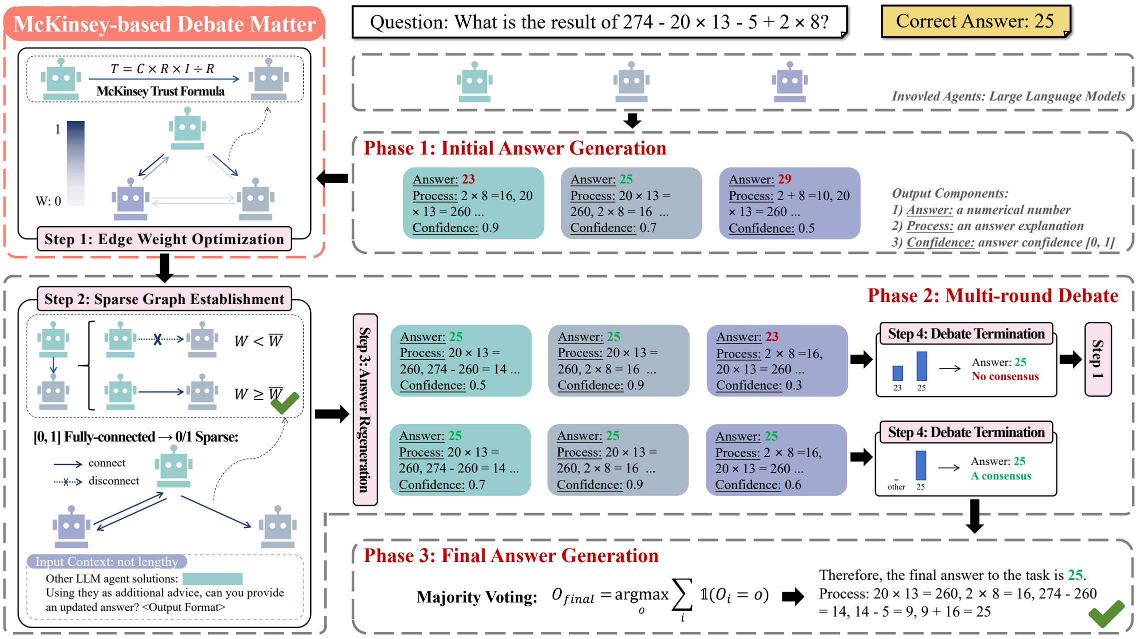

McKinsey-based Debate Matter
$T = C \times R \times I \div R$
McKinsey Trust Formula
1
W: 0
Step 1: Edge Weight Optimization
Step 2: Sparse Graph Establishment
$W < \overline{W}$
$W \geq \overline{W}$
[0, 1] Fully-connected → 0/1 Sparse:
connect
disconnect
Input Context: not lengthy
Other LLM agent solutions:
Using they as additional advice, can you provide an updated answer? <Output Format>
Question: What is the result of 274 - 20 × 13 - 5 + 2 × 8?
Correct Answer: 25
Invovled Agents: Large Language Models
Phase 1: Initial Answer Generation
Answer: 23
Process: 2 × 8 =16, 20 × 13 = 260 ...
Confidence: 0.9
Answer: 25
Process: 20 × 13 = 260, 2 × 8 = 16 ...
Confidence: 0.7
Answer: 29
Process: 2 + 8 =10, 20 × 13 = 260 ...
Confidence: 0.5
Output Components:
1) Answer: a numerical number
2) Process: an answer explanation
3) Confidence: answer confidence [0, 1]
Phase 2: Multi-round Debate
Step 3: Answer Regeneration
Answer: 25
Process: 20 × 13 = 260, 274 - 260 = 14 ...
Confidence: 0.5
Answer: 25
Process: 20 × 13 = 260, 2 × 8 = 16 ...
Confidence: 0.9
Answer: 23
Process: 2 × 8 =16, 20 × 13 = 260 ...
Confidence: 0.3
Step 4: Debate Termination
23
25
Answer: 25
No consensus
Step 1
Answer: 25
Process: 20 × 13 = 260, 274 - 260 = 14 ...
Confidence: 0.7
Answer: 25
Process: 20 × 13 = 260, 2 × 8 = 16 ...
Confidence: 0.9
Answer: 25
Process: 2 × 8 =16, 20 × 13 = 260 ...
Confidence: 0.6
Step 4: Debate Termination
other
25
Answer: 25
A consensus
Phase 3: Final Answer Generation
Majority Voting: $O_{final} = \underset{o}{\mathrm{argmax}} \sum_i \mathbb{1}(O_i = o)$
Therefore, the final answer to the task is 25.
Process: 20 × 13 = 260, 2 × 8 = 16, 274 - 260 = 14, 14 - 5 = 9, 9 + 16 = 25

Figure 2: Overview of our proposed CortexDebate, which is inspired by the working mode of human brain cortex and consists of three phases: (a) Initial Answer Generation: Each LLM agent generates an answer, an explanation, and its confidence score. (b) Multi-round Debate: Participating LLM agents engage in debates guided by a sparse debating graph which is dynamically optimized by MDM module. (c) Final Answer Generation: After multi-round debates, the final answer is generated by majority voting.

the performance reliability on the current question. Intimacy represents the average degree of difference in viewpoints between A_i and A_j in history debates, as the collision of different viewpoints can enhance the debating effectiveness (Xiong et al., 2023). Self-orientation represents the participation level of A_i in the debate (a lower participation level indicates higher self-orientation).

4 Methodology

In this section, we introduce the overall framework of our CortexDebate. As shown in Figure 2 and Algorithm 1, CortexDebate operates in three phases, including *initial answer generation*, *multi-round debate*, and *final answer generation*. Unlike existing MAD methods that establish fully-connected and fixed graphs among LLM agents, our CortexDebate establishes a sparse and dynamic graph, where each LLM agent selectively debates with those that can contribute to its improvement. Besides, CortexDebate evaluates the performance of LLM agents and their usefulness to their debating components, enabling credible graph optimization.

4.1 Phase 1: Initial Answer Generation

When given a problem Q, CortexDebate allows each LLM agent A_i to independently generate an initial output O_i^0 and a self-confidence score H_i^0 (see Appendix E for the specific prompt). To mitigate overconfidence, CortexDebate adopts a recalibration strategy, which has been proven to be effective in prior works (Chen et al., 2023). Our strategy can be expressed as:

$$H_i^0 = \begin{cases} 0.8, & H_i^0 \geq 0.8 \\ 0.6, & 0.6 \leq H_i^0 < 0.8 \\ H_i^0, & 0.3 \leq H_i^0 < 0.6 \\ 0.3, & H_i^0 < 0.3 \end{cases}. \quad (2)$$

4.2 Phase 2: Multi-round Debate

CortexDebate then comes into a debate phase, where the set of agents $\{A_i\}$ engage in D rounds of debate. In the d-th debating round, CortexDebate comprises four steps, including *edge weight optimization*, *sparse graph establishment*, *answer regeneration*, and *debate termination*.

Step 1: Edge Weight Optimization. As the description of Equation (1), MDM calculates the edge weights based on four aspects, including *credibility*, *reliability*, *intimacy*, and *self-orientation*. Following the definition for each aspect in the context of MAD in Section 3.2, the specific calculation of each aspect will be given next.

For $E_{i \to j}$, since the scaling law for LLM agents (Hoffmann et al., 2022) can evaluate abilities of one LLM agent, we use it to calculate credibility C_d, which can be expressed as:

$$\mathcal{L}\left(N, M\right) = \frac{406.4}{N^{0.34}} + \frac{410.7}{M^{0.28}} + 1.69, \quad (3)$$

where N, M, and $\mathcal{L}$ denote the parameter number, the token number of pre-training data, and the pre-training loss of one model, respectively. A smaller loss value indicates better model abilities, and thus C_d is expressed as:

$$C_d = \frac{1}{\mathcal{L}\left(N, M\right)}. \quad (4)$$

For reliability R_d which represents the average confidence score of A_i in its own answers in the preceding $d-1$ rounds, its calculation can be expressed as:

$$R_d = \frac{R_{d-1} \times (d-1) + H_i^{d-1}}{d}. \quad (5)$$

For intimacy I_d, which represents the average degree of difference in viewpoints between A_i and A_j in the preceding $d-1$ rounds, MDM first uses cosine similarity to calculate the textual similarity between O_i^{d-1} and O_j^{d-1}. Subsequently, CortexDebate calculates the average viewpoint similarity between A_i and A_j in the preceding $d-1$ rounds, *i.e.*, $\overline{Sim}_d$, as:

$$\overline{Sim}_d = \frac{\overline{Sim}_{d-1} \times (d-1) + cos(O_i^{d-1}, O_j^{d-1})}{d}, \quad (6)$$

where $cos(a, b)$ calculates cosine similarity between a and b. Since I_d represents the average degree of difference, it is calculated as:

$$I_d = 1 - \overline{Sim}_d. \quad (7)$$

For self-orientation S_d, based on the fact that less group participation indicates that one is more selfish, the MDM module uses the number of times that A_i has debated with other LLM agents in the preceding $d-1$ rounds, denoted as P_d, to indirectly reflect self-orientation. The calculation can be expressed as:

$$S_d = (d-1) \times (n-1) - P_d, \quad (8)$$

where $(d-1) \times (n-1)$ denotes the maximum number of times that one LLM agent can debate with others in the preceding $d-1$ rounds.

Therefore, following Equation (1), the weight of edge $E_{i \to j}$ can be calculated as:

$$W_{i \to j}^d = \frac{C_d \times R_d \times I_d}{S_d}. \quad (9)$$

Step 2: Sparse Graph Establishment. For A_j, it can debate with the other $n-1$ LLM agents. In other words, there are $n-1$ directed edges pointing to it, with A_j as the tail node. CortexDebate determines the set of debating opponents for A_j according to the weights of these edges $\left\{W_{i \to j}^d\right\}_{i=1,\ i \neq j}^{n}$.

Firstly, the average weight of these edges, *i.e.*, $\overline{W}_j^d$, is calculated as:

$$\overline{W}_j^d = \frac{1}{n-1} \sum\nolimits_{i(i \neq j)} W_{i \to j}^d. \quad (10)$$

Secondly, the edges with weights below $\overline{W_j^d}$ are removed, resulting in a sparse debating graph. The process can be expressed as:

$$W_{i \to j}^d = \begin{cases} 1, & W_{i \to j} \geq \overline{W}_j^d \\ 0, & W_{i \to j} < \overline{W}_j^d \end{cases}. \quad (11)$$

Therefore, the debating opponents for A_j, denoted as Deb_j, can be expressed as:

$$Deb_j^d = \left\{A_i \mid W_{i \to j}^d = 1, i \neq j\right\}. \quad (12)$$

Step 3: Answer Regeneration. For LLM agent A_j, it receives the answers of the LLM agents in Deb_j^d, which are generated in the $(d-1)$-th debating round. Afterwards, A_j needs to read and scrutinize these answers, and generate its new answer O_j^d and self-confidence score H_j^d. The input prompt can be expressed as:

$$Prompt_j^d = \left[Ins, Q, \left\{O_k^{d-1}\right\}\right], \quad (13)$$

where Ins denotes the instruction that stimulates A_j to regenerate its answer and $\left\{O_k^{d-1}\right\}$ denotes the set of answers that A_j receives. The specific prompt is shown in Appendix E.

Step 4: Debate Termination. After all the LLM agents have generated their answers, CortexDebate checks whether all the LLM agents reach a consensus (*i.e.*, all the LLM agents agree on the same answer) or the debate reaches the maximum rounds. If so, the whole debating process concludes immediately.

4.3 Phase 3: Final Answer Generation

Once the entire debating process concludes, CortexDebate generates the final answer to the question by majority voting among all the answers generated in the last debating round, which can be expressed as:

$$O_{final} = \arg\max_{o} \sum_{i} \mathbb{1}\left(O_i = o\right), \quad (14)$$

where o denotes a distinct answer generated by any of the LLM agents.

5 Experiments

This section introduces the experimental setup, experimental results, and analysis of our experiments.

Type	Method	GSM-IC	MATH	MMLU	MMLU-pro	GPQA	ARC-C	LongBench	SQuAD
		RA ↑						M-Avg ↑	EM ↑
No Debate	MaV	70.33±1.56	46.00±2.67	69.33±0.22	46.00±4.67	27.33±2.89	76.00±0.67	45.11±1.09	85.33±1.56
Full Debate	MLD	72.67±0.22	47.33±0.89	71.33±1.56	47.33±0.89	28.33±2.89	79.33±0.22	48.87±2.21	86.33±0.22
	RECONCILE	75.67±0.22	50.33±4.22	75.00±2.67	53.67±2.89	31.00±0.67	83.67±2.89	52.55±2.68	88.33±6.89
	ChatEval	74.33±0.89	49.00±0.67	73.00±0.67	49.33±0.89	31.33±0.89	82.67±1.56	53.56±6.16	87.33±6.22
	PRD	77.00±0.67	51.33±0.89	77.33±1.56	54.00±0.67	32.00±2.00	84.33±0.89	50.21±6.09	87.67±4.22
Part Debate	GD	76.00±2.67	49.67±1.56	74.00±2.67	51.67±0.89	32.67±0.22	82.00±2.00	55.97±0.59	90.33±0.89
	ND	73.67±1.56	49.00±0.67	71.67±2.89	48.67±1.56	32.33±1.56	81.33±2.89	54.55±6.18	88.33±1.56
	Ours	**79.33±0.22**	**56.00±0.67**	**82.33±0.22**	**59.33±0.22**	**36.33±1.56**	**88.33±0.89**	**60.31±0.32**	**93.33±0.89**

Table 1: Comparison results on the four different types of tasks. The unit of all the results is "%". The format of the results is "(average result)±(variance)". "↑" means that higher values are better. The best records under each metric are highlighted in bold.

5.1 Experimental Setup

In this part, we introduce the details of the experimental setup.

Tasks. In our experiments, we consider four typical tasks, namely: (a) math task, (b) world knowledge question answering task, (c) reasoning task, and (d) long-context understanding task. For the math task, we use GSM-IC (Shi et al., 2023) and MATH (Hendrycks et al., 2021) datasets. For the world knowledge question answering task, we use MMLU (Hendrycks et al., 2020) and MMLU-pro (Wang et al., 2024) datasets. For the reasoning task, we use GPQA (Rein et al., 2023) and ARC-C (Clark et al., 2018) dataset. For the long-context understanding task, we use LongBench (Bai et al., 2023) and SQuAD (Rajpurkar, 2016) datasets. More details on the employed datasets for experiments can be found in Appendix A.

Evaluation Metrics. For LongBench dataset, we follow (Bai et al., 2023) and utilize the *Macro-Average* (M-Avg), which calculates the average score over major sub-task categories. For SQuAD dataset, we follow (Rajpurkar, 2016) and utilize the *Exact Match* (EM), which calculates the percentage of outputs containing correct answers. For the remaining six datasets, we follow (Shi et al., 2023; Hendrycks et al., 2021, 2020; Wang et al., 2024; Rein et al., 2023; Clark et al., 2018) and utilize the *Result Accuracy* (RA), which calculates the percentage of correct results.

Baseline Methods. Our proposed CortexDebate is compared with the three categories of methods: 1) **No debate:** Multi-agent Voting (MaV) (Wang et al., 2022), 2) **Full debate:** Multi-LLM Debate (MLD) (Du et al., 2023), RECONCILE (Chen et al., 2023), ChatEval (Chan et al., 2023), and Peer Review Debate (PRD) (Xu et al., 2023b), 3) **Part debate:** GroupDebate (GD) (Liu et al., 2024b) and Neighbor Debate (ND) (Li et al., 2024b). Among them, no debate methods are the multi-agent methods without using debating strategies, full debate methods are the MAD methods where each LLM agents are required to debate with all others, and part debate methods are the MAD methods where each LLM agents only debates with part of the others. Detailed introduction of these baseline methods can be found in Appendix F.

For fairness, the maximum number of debating rounds is set to 5 for all debating methods.

Backbone Models. The backbone models involved in the debating system for our experiments are Qwen-2.5-7B-Instruct-Turbo (Team, 2024), Mistral-7B-Instruct (Jiang et al., 2023), Typhoon-1.5-8B-Instruct (Pipatanakul et al., 2023), Llama-3.1-8B-Instruct-Turbo (Dubey et al., 2024), and Gemma-2-9B-Instruct (Yang et al., 2024). For simplicity, we refer to them as Qwen, Mistral, Typhoon, Llama, and Gemma, respectively.

Implementation Details. We follow prior works (Du et al., 2023; Chen et al., 2023; Besta et al., 2024) to experiment on a subset of 100 examples for each dataset. For each experiment, we conduct three runs on the same examples with the same setups and report average results along with their variances. We also conduct large-scale experiments on the more challenging datasets from each task (*i.e.* MATH, MMLU-pro, GPQA, and LongBench) and observe similar results, which are detailed in Appendix C.

5.2 Main Results

In this part, we present the experimental results and detailed analysis to highlight the effectiveness of our proposed CortexDebate.

CortexDebate outperforms baseline methods. Table 1 reports the accuracy of our CortexDebate

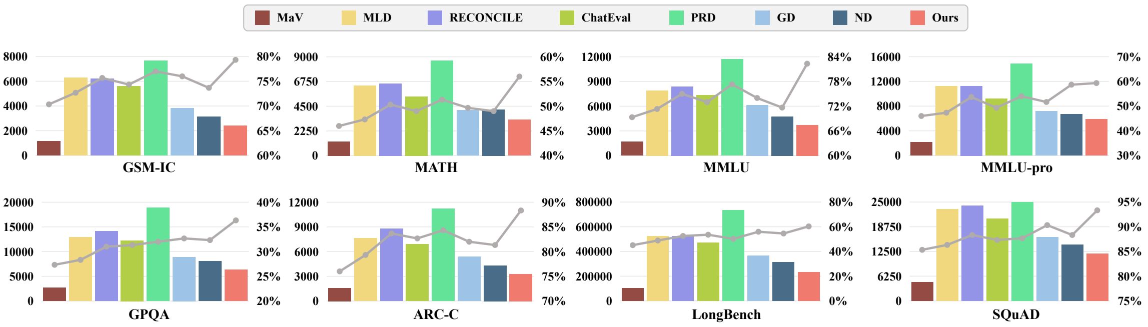
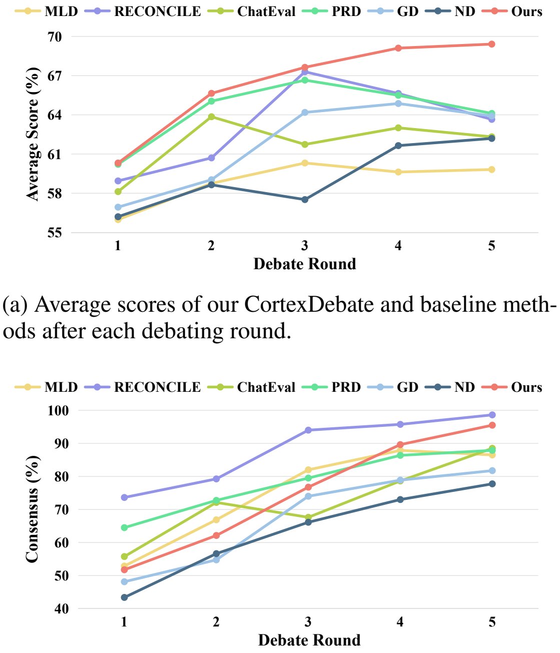


Figure 3: Comparison results of average input context length on eight datasets. We reflect the length of one input context through its token number. In each combined chart, the left vertical axis (representing token number) corresponds to the bar chart, while the right vertical axis (representing task accuracy) corresponds to the line chart.

and baseline methods on eight datasets. Compared with the baseline methods, our CortexDebate achieves the highest accuracy and performs stably on all adopted datasets. Besides, we can find that the effectiveness and stability of the full debate methods (*i.e.*, MLD, RECONCILE, ChatEval, and PRD) drops on complex reasoning and long-context tasks (*i.e.*, GPQA, LongBench, and SQuAD). It is because the reasoning process increases with the complexity of the task, leading to the lengthy context issue mentioned in Section 1. However, our CortexDebate still performs well and stably due to its sparse debating graph which reduces input context length and MDM module which makes each LLM agent debate with those that are helpful to it.

CortexDebate significantly reduces input context length. For each adopted dataset, we calculate the average token number of context input to a single LLM agent in each method and present the results in Figure 3. Compared with MaV, MAD methods generally incur long context input to each LLM agent, indicating a significant challenge in reducing input context length while maintaining superior accuracy in MAD methods. Our proposed CortexDebate takes a further step, as it achieves both shorter input context length and higher task performance compared with other MAD baseline methods. The specific numerical values of the results shown in Figure 3 are presented in Appendix B.

CortexDebate debates effectively and equally. Engaging in more effective debates is what MAD systems strive for. To study this, in Figures 4a and 4b, we plot the average scores and proportion of examples achieving consensus on the answers on eight adopted datasets after each debating round, respectively. From Figure 4a, we have two important observations: (a) As the debate proceeds, the performance of our CortexDebate continues to improve. (b) Compared with the baseline methods, our CortexDebate maintains superior performance and achieves the highest score of 69.41%. From Figure 4b, our observations are likewise twofold: (a) In the initial rounds, since CortexDebate encourages the equal collision of different viewpoints, its consensus proportion is relatively low. However, as the debate proceeds, a high consensus proportion is achieved. (b) Compared with other methods, RECONCILE maintains the highest consensus proportion while its score fluctuates as shown in Figure 4a. This is due to the overconfidence-caused unequal debate, where the debate is dominated by



(a) Average scores of our CortexDebate and baseline methods after each debating round.

(b) Proportion of examples achieving consensus on the answers.

Figure 4: Results of average task scores and consensus proportions for MAD methods after each round.

Method	Score (%)
Fully-connected Graph	60.49
+ MDM	63.76
Sparse Graph	62.72
+ Self-evaluation (RECONCILE)	62.13
+ Peer Evaluation (PRD)	66.71
+ MDM (w/o I_d and S_d in Equation (9))	66.69
+ MDM (Ours)	**69.41**

Table 2: Ablation study on our proposed CortexDebate.

a few LLM agents and others tend to surrender. Differently, our CortexDebate alleviates this issue and maintains equally debates among LLM agents, thereby achieving consistent growth in score and consensus proportion. The numerical results are presented in Appendix B.

5.3 Performance Investigation

In this section, we conduct in-depth investigation on our CortexDebate to analyze its effectiveness. For each method, we use its average score on eight adopted datasets to represent its performance.

Each component of CortexDebate is indispensable. To show that every component of CortexDebate (*i.e.*, sparse debating graph and MDM module) is indispensable, we conduct an ablation study. For the fully-connected graph, we follow the basic MAD framework where each LLM agent debates with all others. For the sparse graph, we use different evaluation strategies to optimize the edge weights of the debating graph, including self-evaluation (Chen et al., 2023), peer evaluation (Xu et al., 2023b), MDM (w/o I_d and S_d in Equation (9)), and MDM (see Appendix D for detailed introduction). As shown in Table 2, compared with “fully-connected graph + MDM”, “spare graph + MDM” increases the average score by 5.65%. It is because sparse debating graph structure alleviates lengthy input context issue and allows LLM agents to make full use of their input information. For different optimization strategies, the average task score of self-evaluation is only 62.13%. It is due to the overconfidence dilemma mentioned in Section 1. Peer evaluation and MDM (w/o I_d and S_d in Equation (9)) alleviate this issue, achieving better performance compared with self-evaluation. Moreover, MDM further improves the task performance, since it considers both the performance of each LLM agent and the usefulness to its debating components, thereby conducting more credible evaluations compared with Peer evaluation and MDM (w/o I_d and S_d in Equation (9)) which only evaluate individual performance.

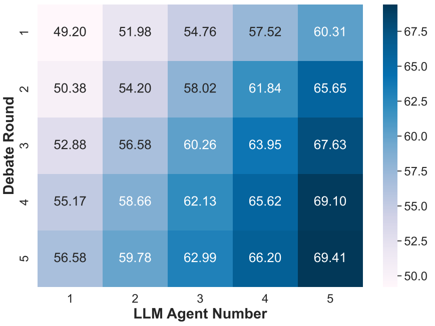


Figure 5: Task performance of CortexDebate under different LLM agent numbers and debating rounds.

CortexDebate excels in large-scale debates. To explore the influence of LLM agent number and debating rounds on our CortexDebate, we evaluate the task performance of CortexDebate under different numbers of participating LLM agents and debating rounds. We present the results in Figure 5. We can see that as the number of LLM agents and the debating rounds increase, the task performance of our CortexDebate continues to improve. Moreover, compared with debating rounds, the increase in the number of LLM agents contributes more to the performance improvement of CortexDebate. These results demonstrate the potential of CortexDebate for application in large-scale debates.

6 Conclusion

In this paper, we propose a new MAD method termed “CortexDebate” to improve the reasoning abilities of multi-agent interaction systems. Specifically, our CortexDebate establishes a sparse debating graph among participating LLM agents, which reduces input information burdens of LLM agents. Besides, by integrating the McKinsey Trust Formula, our proposed MDM module conducts credible evaluations to gradually optimize the debating graph, making the debating process equal, in-depth, and effective. Due to the above designs, our method alleviates two major issues faced by existing MAD systems (*i.e.*, too lengthy input contexts and overconfidence-caused unequal debates), and shows superior performance to various state-of-the-art MAD methods on various typical tasks. In the future, we plan to continue exploring the potential of CortexDebate in large-scale debates and complex tasks (*i.e.*, domain expert systems).

Limitations

Despite the impressive performance of our proposed CortexDebate, we acknowledge that it has two main limitations. Firstly, as a multi-agent debate method, compared with single-agent methods, it is inevitable that there will be a decrease in efficiency and an increase in cost when solving tasks. Secondly, despite the success, the reasoning ability of LLM agents remains an important factor that limits the performance of CortexDebate. Although our proposed CortexDebate improves the debate strategy among LLM agents, mistakes may still occur due to the poor reasoning ability of LLM agents.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. In *Annual Meeting of the Association for Computational Linguistics*.

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 *AAAI Conference on Artificial Intelligence*.

Jonah Brown-Cohen, Geoffrey Irving, and Georgios Piliouras. 2023. Scalable ai safety via doubly-efficient debate. In *International Conference on Machine Learning*.

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. In *International Conference on Learning Representations*.

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2023. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. *arXiv preprint arXiv:2309.13007*.

Steffi Chern, Ethan Chern, Graham Neubig, and Pengfei Liu. 2024. Can large language models be trusted for evaluation? scalable meta-evaluation of llms as evaluators via agent debate. *arXiv preprint arXiv:2401.16788*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

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. In *International Conference on Machine Learning*.

Zhihua Duan and Jialin Wang. 2024. Enhancing multi-agent consensus through third-party llm integration: Analyzing uncertainty and mitigating hallucinations in large language models. *arXiv preprint arXiv:2411.16189*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yong Guan, Hao Peng, Lei Hou, and Juanzi Li. 2025. Mmd-ere: Multi-agent multi-sided debate for event relation extraction. In *Proceedings of the 31st International Conference on Computational Linguistics*.

Zhitao He, Pengfei Cao, Chenhao Wang, Zhuoran Jin, Yubo Chen, Jiexin Xu, Huaijun Li, Kang Liu, and Jun Zhao. 2024. Agentscourt: Building judicial decision-making agents with court debate simulation and legal knowledge augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*.

Mahmood Hegazy. 2024. Diversity of thought elicits stronger reasoning capabilities in multi-agent debate frameworks. *arXiv preprint arXiv:2410.12853*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

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. *Advances in Neural Information Processing Systems*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *Advances in Neural Information Processing Systems*.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. 2024. Routerbench: A benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*.

Zhe Hu, Hou Pong Chan, Jing Li, and Yu Yin. 2025. Debate-to-write: A persona-driven multi-agent framework for diverse argument generation. In *International Conference on Computational Linguistics*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R Bowman, Tim Rocktäschel, and Ethan Perez. 2024. Debating with more persuasive llms leads to more truthful answers. In *International Conference on Machine Learning*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*.

Eric Lamarre, T Mansour, and J Tetrault. 2012. Mckinsey on cooperatives.

Renhao Li, Minghuan Tan, Derek F Wong, and Min Yang. 2024a. Coevol: Constructing better responses for instruction finetuning through multi-agent cooperation. In *Conference on Empirical Methods in Natural Language Processing*.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. 2024b. Improving multi-agent debate with sparse communication topology. In *Findings of the Association for Computational Linguistics: EMNLP 2024*.

Jingcong Liang, Rong Ye, Meng Han, Ruofei Lai, Xinyu Zhang, Xuanjing Huang, and Zhongyu Wei. 2024. Debatrix: Multi-dimensinal debate judge with iterative chronological analysis based on llm. In *Findings of the Association for Computational Linguistics: ACL 2024*.

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. In *Annual Meeting Of The Association For Computational Linguistics*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

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

Julian Michael, Salsabila Mahdi, David Rein, Jackson Petty, Julien Dirani, Vishakh Padmakumar, and Samuel R Bowman. 2023. Debate helps supervise unreliable experts. *arXiv preprint arXiv:2311.08702*.

Chau Pham, Boyi Liu, Yingxiang Yang, Zhengyu Chen, Tianyi Liu, Jianbo Yuan, Bryan A Plummer, Zhaoran Wang, and Hongxia Yang. 2023. Let models speak ciphers: Multiagent debate through embeddings. In *International Conference on Learning Representations*.

Kunat Pipatanakul, Phatrasek Jirabovonvisut, Potsawee Manakul, Sittipong Sripaisarnmongkol, Ruangsak Patomwong, Pathomporn Chokchainant, and Kasima Tharnpipitchai. 2023. Typhoon: Thai large language models. *arXiv preprint arXiv:2312.13951*.

P Rajpurkar. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing*.

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 q&a benchmark. *arXiv preprint arXiv:2311.12022*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*.

Qiushi Sun, Zhangyue Yin, Xiang Li, Zhiyong Wu, Xipeng Qiu, and Lingpeng Kong. 2024a. Corex: Pushing the boundaries of complex reasoning through multi-model collaboration. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Xiaoxi Sun, Jinpeng Li, Yan Zhong, Dongyan Zhao, and Rui Yan. 2024b. Towards detecting llms hallucination via markov chain-based multi-agent debate framework. *arXiv preprint arXiv:2406.03075*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Michel Thiebaut de Schotten and Stephanie J Forkel. 2022. The emergent properties of the connected brain. *Science*, 378(6619):505–510.

Lifu Tu, Semih Yavuz, Jin Qu, Jiacheng Xu, Rui Meng, Caiming Xiong, and Yingbo Zhou. 2023. Unlocking anticipatory text generation: A constrained approach for faithful decoding with large language models. In *Conference on Empirical Methods in Natural Language Processing*.

Fanqi Wan, Longguang Zhong, Ziyi Yang, Ruijun Chen, and Xiaojun Quan. 2024. Fusechat: Knowledge fusion of chat models. *arXiv preprint arXiv:2408.07990*.

Haotian Wang, Xiyuan Du, Weijiang Yu, Qianglong Chen, Kun Zhu, Zheng Chu, Lian Yan, and Yi Guan. 2025. Learning to break: Knowledge-enhanced reasoning in multi-agent debate system. *Neurocomputing*, 618:129063.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*.

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

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. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023a. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. In *International Conference on Learning Representations*.

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

Ziyi Yang, Fanqi Wan, Longguang Zhong, Tianyuan Shi, and Xiaojun Quan. 2024. Weighted-reward preference optimization for implicit model fusion. *arXiv preprint arXiv:2412.03187*.

Zhangyue Yin, Qiushi Sun, Cheng Chang, Qipeng Guo, Junqi Dai, Xuanjing Huang, and Xipeng Qiu. 2023. Exchange-of-thought: Enhancing large language model capabilities through cross-model communication. In *Conference on Empirical Methods in Natural Language Processing*.

Luke Yoffe, Alfonso Amayuelas, and William Yang Wang. 2024. Debunc: mitigating hallucinations in large language model agent communication with uncertainty estimations. *arXiv preprint arXiv:2407.06426*.

Jintian Zhang, Xin Xu, Ningyu Zhang, Ruibo Liu, Bryan Hooi, and Shumin Deng. 2024a. Exploring collaboration mechanisms for llm agents: A social psychology view. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Mingqing Zhang, Haisong Gong, Qiang Liu, Shu Wu, and Liang Wang. 2024b. Breaking event rumor detection via stance-separated multi-agent debate. *arXiv preprint arXiv:2412.04859*.

A Dataset Details

The eight datasets used in our experiments are classic datasets that are widely employed to evaluate the performance of agent-based methods. Here, we provide an introduction to the eight datasets used in our experiments.

GSM-IC. It is a grade-school math problem dataset derived from GSM8K (Cobbe et al., 2021). For each problem in GSM8K, GSM-IC keeps the base problem description and adds to it one irrelevant sentence that does not affect the solution of the problem.

MATH. It is a math dataset containing challenging competition mathematics problems. Each of them has a full step-by-step solution.

MMLU. It contains 57 types of multiple-choice problems, such as elementary mathematics, US history, computer science, and so on. To acquire high performance on the MMLU dataset, models must possess extensive world knowledge and strong problem-solving ability.

MMLU-pro. It contains questions sourced from multiple origins, such as MMLU, TheoremQA, and SciBench. Moreover, it expands the option number of each problem from 4 to 10.

GPQA. It contains 448 graduate-level question-answering problems, covering knowledge in various fields such as biology, physics, and chemistry.

ARC-C. It contains complex questions on natural science, presented in the form of multiple-choice options.

LongBench. It is a dataset designed to evaluate the long-context understanding capabilities of models. It encompasses six major categories of tasks, including single-document QA, multi-document QA, summarization, few-shot learning, code completion, and synthetic tasks.

SQuAD. It is a dataset used to evaluate the reading comprehension ability of models. The dataset requires models to answer different questions from given long texts.

B Supplementary Experimental Results

In this section, we provide the experimental result data involved in the charts which are presented in Sections 5.2 and 5.3.

For Figure 3, we provide the data in Table 5. Compared with the full debate methods (*i.e.*, MLD, RECONCILE, ChatEval, and MPRC), our CortexDebate significantly reduces the length of the contextual input for each LLM agent, with a maximum reduction of 70.79%. Moreover, compared with the part debate methods (*i.e.*, GD and ND), our CortexDebate can reduce the input context length by at least 17.62%.

Type	Method	MATH	MMLU-pro	GPQA	LongBench
		RA ↑			M-Avg ↑
No Debate	MaV	47.40	46.30	29.10	43.35
Full Debate	MLD	49.20	48.40	30.60	46.26
	RECONCILE	50.70	53.10	30.80	48.33
	ChatEval	49.90	49.30	31.10	51.23
	PRD	51.20	54.20	32.40	47.67
Part Debate	GD	50.30	51.30	34.20	54.58
	ND	49.50	49.10	32.80	54.14
	Ours	**56.30**	**58.90**	**36.60**	**59.63**

Table 3: Comparison results on the four datasets. The unit of all the results is "%". "↑" means that higher values are better. The best records under each metric are highlighted in bold.

Type	Method	MATH	MMLU-pro	GPQA	LongBench
No Debate	MaV	1316.85	2268.10	2567.90	125034.39
Full Debate	MLD	6408.39	11412.75	12868.37	585365.43
	RECONCILE	6723.84	11334.12	14061.76	605688.14
	ChatEval	5571.07	9219.88	12369.62	553114.66
	PRD	8849.70	14946.31	18851.29	815449.95
Part Debate	GD	4217.23	7265.23	8817.46	447987.68
	ND	4175.27	6673.75	7971.60	394905.47
	Ours	**3355.20**	**6001.33**	**6503.76**	**321109.57**

Table 4: Comparison results of average input context length on adopted datasets. Each result represents the average token number of input context. The results in gray indicate that they are not included in result comparison, since their corresponding method (MaV) is not MAD method. The best records among the MAD methods on each dataset are highlighted in bold.

Type	Method	GSM-IC	MATH	MMLU	MMLU-pro	GPQA	ARC-C	LongBench	SQuAD
No Debate	MaV	1161.18	1277.73	1670.48	2144.39	2653.92	1582.67	105020.51	4724.95
Full Debate	MLD	6287.39	6397.97	7905.84	11213.07	12947.80	7605.01	525177.19	23185.66
	RECONCILE	6196.09	6574.29	8409.61	11260.15	14107.64	8770.85	525765.69	24112.64
	ChatEval	5600.22	5394.71	7325.20	9160.49	12252.56	6918.96	473070.19	20781.94
	PRD	7651.62	8652.23	11691.83	14829.46	18837.51	11232.72	735370.99	33350.02
Part Debate	GD	3828.95	4139.04	6156.13	7159.45	8906.51	5381.72	367835.88	16073.50
	ND	3149.04	4207.18	4740.46	6647.27	8016.54	4311.26	314993.80	14218.49
	Ours	**2413.35**	**3262.79**	**3727.65**	**5897.94**	**6340.26**	**3280.71**	**230956.05**	**11965.81**

Table 5: Comparison results of average input context length on eight datasets. Each result represents the average token number of input context. The results in gray indicate that they are not included in result comparison, since their corresponding method (MaV) is not MAD method. The best records among the MAD methods on each dataset are highlighted in bold.

Type	Method	Score (%)					Consensus (%)				
		1	2	3	4	5	1	2	3	4	5
Full Debate	MLD	55.99	58.76	60.32	59.63	59.82	52.88	66.88	82.00	87.88	86.50
	RECONCILE	58.95	60.71	67.30	65.64	63.65	73.63	79.25	94.00	95.75	98.63
	ChatEval	58.13	63.86	61.74	63.01	62.32	55.75	72.13	67.63	78.63	88.50
	PRD	60.22	65.05	66.66	65.50	64.11	64.50	72.75	79.50	86.38	87.88
Part Debate	GD	56.94	59.04	64.19	64.87	63.91	48.13	54.75	74.00	78.88	81.75
	ND	56.21	58.65	57.52	61.65	62.19	43.38	56.63	66.13	73.00	77.75
	Ours	60.31	65.65	67.63	69.10	69.41	51.75	62.13	76.75	89.63	95.50

Table 6: Experimental results of average task scores and consensus proportion for all the methods after each debate round.

For Figure 4, we provide the data in Table 6. After each debating round, our CortexDebate achieves the highest average score compared with the baseline methods, with a global maximal score of 69.41% (67.30% for the baseline methods).

C Additional Experiments

In this section, we present large-scale experiments of our CortexDebate and baseline methods to further demonstrate the superiority of CortexDebate.

Experimental Setup. For each task (*i.e.*, math, world knowledge question answering, reasoning, and long-context understanding), we conduct experiments on the more challenging one of the two datasets (*i.e.*, MATH, MMLU-pro, GPQA, and LongBench). For each adopted dataset, we experiment on a subset of 1000 examples. Besides, the backbone models and evaluation metrics used in the experiments are the same as mentioned in Section 5.1.

Results. The experimental results are presented in Table 3. Consistent with the experimental results reported in Table 1, our proposed CortexDebate achieves the best performance on all the datasets compared with baseline methods. For instance, CortexDebate achieves a maximal RA of 56.30% on MATH dataset, 58.90% on MMLU-pro dataset, 36.60% on GPQA dataset, and 59.63% on LongBench dataset, respectively. Moreover, for each method, we calculate the average token numbers of the contexts input to one LLM agent on each dataset and present the results in Table 4. Compared with the full debate methods (*i.e.*, MLD, RECONCILE, ChatEval, and MPRC), our CortexDebate significantly reduces the length of the contextual input for each LLM agent, with a maximum reduction of 65.50%. Moreover, compared with the part debate methods (*i.e.*, GD and ND), our CortexDebate can reduce the input context length by at least 17.40%.

D Introduction of Evaluation Strategies

Here we introduce the details of the evaluation strategies (*i.e.*, self-evaluation, peer evaluation, part MDM) mentioned in Section 5.3.

Self-evaluation. In this strategy, each LLM agent is required to generate a confidence score for its generated answer. Each LLM agent will only debate with the LLM agents whose confidence scores are above the average of the entire graph.

Peer Evaluation. For each LLM agent, its answer is scored by other LLM agents, and the final score of the answer is the average of the received scores. Each LLM agent will only debate with the LLM agents whose scores are above the average of the entire graph.

MDM (w/o I_d and S_d in Equation (9)). For McKinsey Trust Formula used in MDM module, this strategy only considers the first two aspects (*i.e.*, credibility and reliability) which evaluate individual abilities, neglecting the last two aspects (*i.e.*, intimacy and self-orientation) which evaluate the debate effectiveness between two LLM agents.

E Prompts in CortexDebate

We provide the specific prompts of our proposed CortexDebate in Table 7. For initial answer generation, CortexDebate follows (Kojima et al., 2022) and prompts each LLM agent to solve the problem step by step. For answer regeneration, the prompt contains three parts: (a) An instruction that stimulates LLM agents to generate their new answers and self-confidence scores after scrutinizing other answers. (b) A description of the problem. (c)

Type	Prompt
Initial Answer Generation	Question: {the description of the question} Please think it step by step and generate an answer and an explanation for your answer. Also, evaluate how confident you are that your answer is correct. Your confidence score should between 0 and 1. The format of your answer must be: Answer: (...) Explanation: (...) Confidence Score: (...)
Answer Regeneration	Question: {the description of the question} There are some answers generated by other LLM agents: One LLM agent answer: {answer} One LLM agent answer: {answer} ... ... Using these answers as additional information, please generate a new answer and an explanation for your answer. Also, evaluate how confident you are that your answer is correct. Your confidence score should between 0 and 1. The format of your answer must be: Answer: (...) Explanation: (...) Confidence Score: (...)

Table 7: Prompts of our proposed CortexDebate used in the experiments.

Some answers generated by other LLM agents.

F Introduction of Baseline Methods

Here we introduce the details of the baseline methods (*i.e.*, Multi-agent Voting, Multi-LLM Debate, RECONCILE, ChatEval, Peer Review Debate, GroupDebate, and Neighbor Debate) in our experiments.

Multi-agent Voting. This method adopts a majority voting strategy to aggregate responses from multiple LLM agents. Specifically, each LLM agent independently generates a response to the given question. The final prediction is then determined through majority voting.

Multi-LLM Debate. Firstly, each LLM agent generates an answer to the test question. Then, each LLM agent reads and critiques the answers generated by other LLM agents, and generates its new answer. This step is repeated multiple times. After that, the final answer is obtained through majority voting among the answers generated by all the LLM agents in the last round of debate. The specific prompt used in the experiments is shown in Figure 6.

RECONCILE. Given a problem, each LLM agent first generates an answer and its uncertainty for the answer. Then all LLM agents enter a multi-round debate. Each debating round consists of each LLM agent generating a revised answer and its new uncertainty based on the answers generated by all other LLM agents from the previous round. After the multi-round debate, RECONCILE obtains the final answer through majority voting. The specific prompt used in the experiments is shown in Figure 7.

ChatEval. ChatEval uses an extra LLM agent to summarize the debating results in each round of debate. The specific prompt of debating summary used in the experiments is shown in Figure 8. The summary text generated in the current round of debate will be input to each LLM agent as supplementary information in the next round of debate.

Peer Review Debate. Similar to RECONCILE, this method also evaluates all the answers in each round of debate. However, instead of self-evaluation, this method employs a peer review

Debating Prompt for Each LLM Agent

These are the solutions to the problem from other agents: [other answers]
Using the opinion of other LLM agents as additional advice, can you give an updated response ...

Figure 6: Prompt of Multi-LLM Debate used in the experiments.

Initial Answer Generation

{convincing_samples}
Q: {test_question}
Please answer the question with step-by-step reasoning.
Also, evaluate your confidence level (between 0.0 and 1.0) to indicate the possibility of your answer being right.

Debate

{convincing_samples}
{initial_prompt}
Carefully review the following solutions from other agents as additional information, and provide your own answer and step-by-step reasoning to the question.

Clearly state which point of view you agree or disagree with and why.

There are {majority_num} agents think the answer is {majority_ans}.
One agent solution: {agent_reasoning} {agent_ans} {agent_confidence}
One agent solution: {agent_reasoning} {agent_ans} {agent_confidence}

There are {minority_num}agents think the answer is {minority_ans}.
One agent solution: {agent_reasoning} {agent_ans} {agent_confidence}

Figure 7: Prompt of RECONCILE used in the experiments.

strategy where the answer generated by each LLM agent is evaluated by other LLM agents. The specific prompt used in the experiments is shown in Figure 9.

GroupDebate. This method divides all participating LLM agents into several debate groups, with each group conducting internal debates. After the internal debates, the result of each debate group is summarized and placed into a shared pool. After that, each group retrieves the debate summaries of all groups from the pool, which serve as the input for all the LLM agents in the next round. The specific prompt used in the experiments is shown in Figure 10.

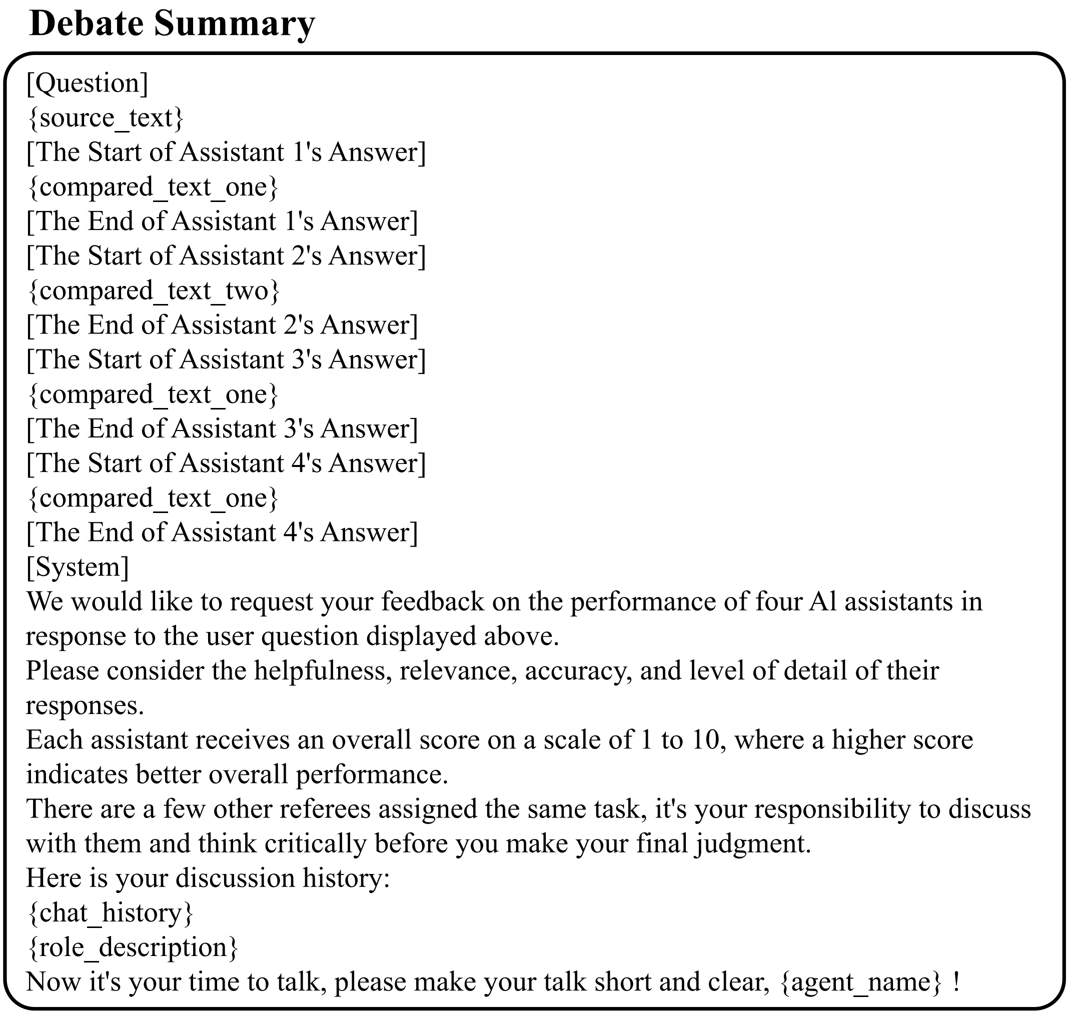
Neighbor Debate. In this method, each LLM agent only debates with its neighbors. The specific prompt used in the experiments is shown in Figure 11.

G CortexDebate Algorithm

In this section, we provide the detailed algorithm of our proposed CortexDebate. As present in Algorithm 1, we strictly follow Sections 3 and 4, and provide the whole execution process of our proposed CortexDebate.

Algorithm 1 CortexDebate Method

```
Input: Number of LLM agents $n$, set of LLM agents $\{A_i\}_{i=1}^{n}$, set of directed edges $\{E_{i\to j}\}$, test question
    $Q$, maximum debating rounds $D$, answer extraction $ans\left(\cdot\right)$
Output: Final answer $O_{final}$
for $i = 1$ to $n$ do
    $O_i^0, H_i^0 \leftarrow A_i\left(Q\right)$                                ▷ Phase 1: Initial Answer Generation
    Recalibration $H_i^0$ based on Equation (2)
    Calculate $C^i$ based on Equations (3) and (4)
    $P_0^i \leftarrow 0$
end for
$O \leftarrow \{O_i^0\}_{i=1}^{n}$                                              ▷ Phase 2: Multi-round Debate
for $d = 1$ to $D$ do
    for $i = 1$ to $n$ do
        Calculate $R_d^i$ and $S_d^i$ based on Equations (5) and (8), respectively
        for $j = 1$ to $n$ do
            if $i \neq j$ then
                Calculate $\overline{Sim}_d$ and $I_d^i$ based on Equations (6) and (7), respectively
                Calculate $W_{i\to j}^d$ based on Equation (9)          ▷ Step 1: Edge Weight Optimization
            end if
        end for
    end for
    for $j = 1$ to $n$ do
        $Deb_j^d, Others_j^d \leftarrow \emptyset$                            ▷ Step 2: Sparse Graph Establishment
        Calculate $\overline{W}_j^d$ based on Equation (10)
        for $i = 1$ to $n$ do
            if $i \neq j$ then
                Calculate $W_{i\to j}^d$ based on Equation (11)
                if $W_{i\to j}^d = 1$ then
                    $Deb_j^d \leftarrow Deb_j^d \cup \{A_i\}$
                    $Others_j^d \leftarrow Others_j^d \cup \left\{O_i^{d-1}\right\}$
                end if
            end if
        end for
        $O_i^d, H_i^d \leftarrow A_j\left(Q,\ Others_j^d\right)$                   ▷ Step 3: Answer Regeneration
        Recalibration $H_i^d$ based on Equation (2)
    end for
    $is\_end \leftarrow$ True                                                 ▷ Step 4: Debate Termination
    for $i = 2$ to $n$ do
        if $ans\left(O_1^d\right) \neq ans\left(O_i^d\right)$ then
            $is\_end \leftarrow$ False
            break
        end if
    end for
    $O \leftarrow \{O_i^d\}_{i=1}^{n}$
    if $is\_end =$ True then
        break
    end if
end for
$o \leftarrow$set$(O_1,\ O_2,\ \cdots,\ O_n)$
Get $O_{final}$ based on Equation (14)                                 ▷ Phase 3: Final Answer Generation
return $O_{final}$
```



Debate Summary

[Question]
{source_text}
[The Start of Assistant 1's Answer]
{compared_text_one}
[The End of Assistant 1's Answer]
[The Start of Assistant 2's Answer]
{compared_text_two}
[The End of Assistant 2's Answer]
[The Start of Assistant 3's Answer]
{compared_text_one}
[The End of Assistant 3's Answer]
[The Start of Assistant 4's Answer]
{compared_text_one}
[The End of Assistant 4's Answer]
[System]
We would like to request your feedback on the performance of four Al assistants in response to the user question displayed above.
Please consider the helpfulness, relevance, accuracy, and level of detail of their responses.
Each assistant receives an overall score on a scale of 1 to 10, where a higher score indicates better overall performance.
There are a few other referees assigned the same task, it's your responsibility to discuss with them and think critically before you make your final judgment.
Here is your discussion history:
{chat_history}
{role_description}
Now it's your time to talk, please make your talk short and clear, {agent_name} !

Figure 8: Debating summary prompt of ChatEval used in the experiments.

Initial Answer Generation

Can you solve the following problem? {Question}
Explain your reasoning. Your final answer should be in the form \boxed{answer}, at the end of your response.

Peer Review

Here is a solution from another agent: {Answer B}
Please examine this agent's reasoning process step by step and offer feedback on its reasoning.
You can rate your confidence in your feedback on a scale from 1-10, where 10 indicates the highest level of confidence.

Answer Revise

Here are the feedbacks for your solution from other agents:
One agent feedback: {Feedback B → A}
One agent feedback: {Feedback C → A}
One agent feedback: {Feedback D → A}
One agent feedback: {Feedback E → A}
Using other agents' solutions and feedbacks as additional information, can you provide your answer to the math problem?
The original math problem is {Question}
Your final answer should be a single numerical number, in the form \boxed{answer}, at the end of your response.

Figure 9: Prompt of Peer Review Debate used in the experiments.

System

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

Can you solve the following problem? <Problem>
Explain your reasoning. <Output format>.

Intra-group Debate

These are the recent opinions from other agents: <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

These are the recent/updated 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

These are the recent opinions from all groups: Your group response: <group summary>, Other group responses: <other 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>.

Figure 10: Prompt of GroupDebate Debate used in the experiments.

System

You are a helpful assistant. Your task is to assist in solving a problem by providing a clear and detailed solution. Your final answer should be in the form of {{answer}}, at the end of your response.

Initial Answer Generation

Can you solve the following problem? {question}
Explain your reasoning. Your final answer should be in the form of {{answer}}, at the end of your response.

Debate

These are the solutions to the problem from other agents:
One agent solution: {reference solution}
One agent solution: {reference solution}
One agent solution: {reference solution}
One agent solution: {reference solution}
Using the solutions from other agents as additional information, can you provide your answer to the problem? The original problem is {question}. Your final answer should be in the form of {{answer}}, at the end of your response.

Figure 11: Prompt of Neighbor Debate used in the experiments.