

Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate

Tian Liang^{1,3*} Zhiwei He^{2*} Wenxiang Jiao^{3*} Xing Wang^{3†} Yan Wang³

Rui Wang² Yujiu Yang^{1†} Shuming Shi³ Zhaopeng Tu³

¹Tsinghua University ²Shanghai Jiao Tong University ³Tencent AI Lab

¹{liangt21@mails, yang.yujiu@sz}.tsinghua.edu.cn

²zwhe.cs@sjtu.edu.cn ³{joelwxjiao, brightxwang, zptu}@tencent.com

Abstract

Modern large language models (LLMs) like ChatGPT have shown remarkable performance on general language tasks but still struggle on complex reasoning tasks, which drives the research on cognitive behaviors of LLMs to explore human-like problem-solving strategies. Along this direction, one representative strategy is self-reflection, which asks an LLM to refine the solution with the feedback generated by itself iteratively. However, our study shows that such reflection-style methods suffer from the *Degeneration-of-Thought (DoT)* problem: once the LLM has established confidence in its solutions, it is unable to generate novel thoughts later through reflection even if its initial stance is incorrect. To address the DoT problem, we propose a **Multi-Agent Debate (MAD)** framework, in which multiple agents express their arguments in the state of “tit for tat” and a judge manages the debate process to obtain a final solution. Clearly, our MAD framework encourages divergent thinking in LLMs which would be helpful for tasks that require deep levels of contemplation. Experiment results on two challenging datasets, commonsense machine translation and counter-intuitive arithmetic reasoning, demonstrate the effectiveness of our MAD framework. Extensive analyses suggest that the adaptive break of debate and the modest level of “tit for tat” state are required for MAD to obtain good performance. Moreover, we find that LLMs might not be a fair judge if different LLMs are used for agents. Code is available at <https://github.com/Skytliang/Multi-Agents-Debate>.

1 Introduction

Large language models (LLMs) have shown remarkable performance on general language tasks (Jiao et al., 2023; Wu et al., 2023; Bang

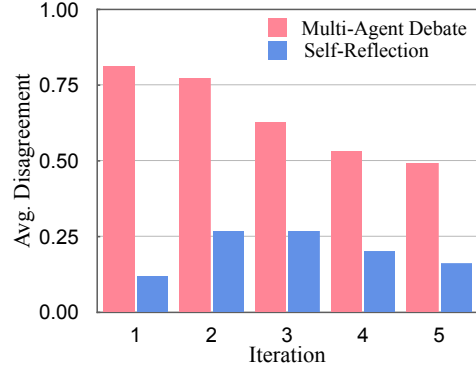


Figure 1: Disagreement between two adjacent iterations with respect to the iteration of debate/self-reflection.

et al., 2023) but still struggle on complex reasoning tasks (Zhu et al., 2023a; Gou et al., 2023), which drives the research on cognitive behaviors of LLMs to explore human-like problem-solving strategies. In particular, *self-reflection* (Madaan et al., 2024; Shinn et al., 2024), a concept that usually refers to the process of introspection and examination of a person’s own thoughts, has been explored to solve intricate tasks that could be challenging for a zero-shot generation or even chain-of-thought (CoT) prompting (Wei et al., 2022). Specifically, self-reflection involves an iterative refinement process such that the LLM generates a new answer based on the answers and feedback in previous iterations and then provides feedback for the new answer. While self-reflection can be effective in creating better answers, it is highly dependent on the self-evaluation capabilities of LLMs, which are not formally guaranteed (Shinn et al., 2024).

In this work, we focus on the **Degeneration-of-Thought (DoT)** problem in self-reflection, which is proposed and defined by us for the first time. Formally, DoT describes the following scenario:

Once the LLM-based agent has established confidence in its answers, it is unable to generate novel thoughts later through self-reflection even if the initial stance is incorrect.

* Contributed equally. Work was done when Tian and Zhiwei were interning at Tencent AI Lab.

† Xing and Yujiu are co-corresponding authors.

To demonstrate this problem, we force the agents to engage in a debate or self-reflection for 5 rounds before reaching an answer. Next, we manually determine the disagreement as 1 and agreement as 0 between two adjacent iterations. We define the average disagreement in iteration i as the percentage of opposition occurring between two debaters across multiple debates (or self-confliction in self-reflection). We show the trends in Figure 1. The low disagreement of self-reflection suggests that the LLM sticks to the incorrect answers predicted by CoT and is unable to engage in meaningful self-reflection.

There are various factors (Bortolotti, 2011; Keestra, 2017) that could result in DoT, and we outline three here: (1) *Bias and Distorted Perception*. Self-perception can be influenced by biases, preconceived notions, and distorted thinking patterns, which can be learned from the massive amount of data during pretraining. If an LLM’s self-reflection is clouded by such biases or distorted thinking, it can lead to inaccurate conclusions instinctively. (2) *Rigidity and Resistance to Change*. Self-reflection often involves challenging one’s beliefs, assumptions, and behaviors. If an LLM is resistant to change or holds rigid beliefs, it may struggle to engage in meaningful self-reflection that leads to better answers. (3) *Limited External Feedback*. Self-reflection is primarily an internal process, but external feedback can provide valuable perspectives and insights. Without considering external feedback, an LLM may miss important blind spots or alternative viewpoints that can enrich its self-reflection.

To address the DoT issue, we leverage another fundamental characteristic of human problem-solving, i.e., debate, to encourage divergent thinking in LLMs. Specifically, we propose the **MAD** framework, short for **Multi-Agent Debate**, where two agents express their own arguments in the state of “tit for tat” and a judge monitors and manages the debate process to obtain a final solution. The nature of MAD determines that (1) The distorted thinking of one agent can be corrected by the others; (2) The resistance to change of one agent will be complemented by the others; and (3) each agent can obtain external feedback from the others. Therefore, MAD is less susceptible to the factors of DoT, and can explore divergent chain-of-thoughts to achieve accurate answers.

We conducted experiments on both natural language generation and understanding through two challenging tasks, namely, Commonsense Machine

Translation (Common MT) and Counter-Intuitive Arithmetic Reasoning (Counter-Intuitive AR). The common characteristic of the two tasks is that our instincts are mostly incorrect based on only the superficial expressions of the questions, and deeper levels of contemplation are required for better answers. Experimental results demonstrate that our MAD framework outperforms the baseline methods, especially, GPT-3.5-Turbo with MAD can surpass the performance of GPT-4 on Common MT.

The contributions of this work are summarized as follows:

- We propose and define the Degeneration-of-Thought (DoT) problem in self-reflection, and address it by proposing the Multi-Agent Debate (MAD) framework to explore divergent chain-of-thoughts.
- We demonstrate the effectiveness of MAD on two challenging tasks, and find that GPT-3.5-Turbo with MAD can even surpass GPT-4 on the Common MT dataset.
- Experimental results show that the adaptive break strategy and the modest level of “tit for tat” state are required for performance improvement. In addition, we find that the llm-based judge shows a preference to the side with the same LLM as the backbone.

2 Multi-Agent Debate Framework

Figure 2 illustrates the general framework of MAD, where two debaters and a judge are involved in a debate to resolve a math problem while self-reflection descends into the trap of DoT. Generally, our MAD framework is composed of three components which are elaborated as follows:

Meta Prompts. We use meta prompts to introduce the topic to be solved, the number of debaters, the iteration limit, and other requirements. An example of meta prompts for the arithmetic reasoning task in Figure 2 is:

You are a debater. Hello and welcome to the debate competition. It’s not necessary to fully agree with each other’s perspectives, as our objective is to find the correct answer. The debate topic is stated as follows: <debate topic>.

As seen, we require the agents to “tit for tat” (e.g. contents underlined in meta prompts above) so as to create an atmosphere of debate.

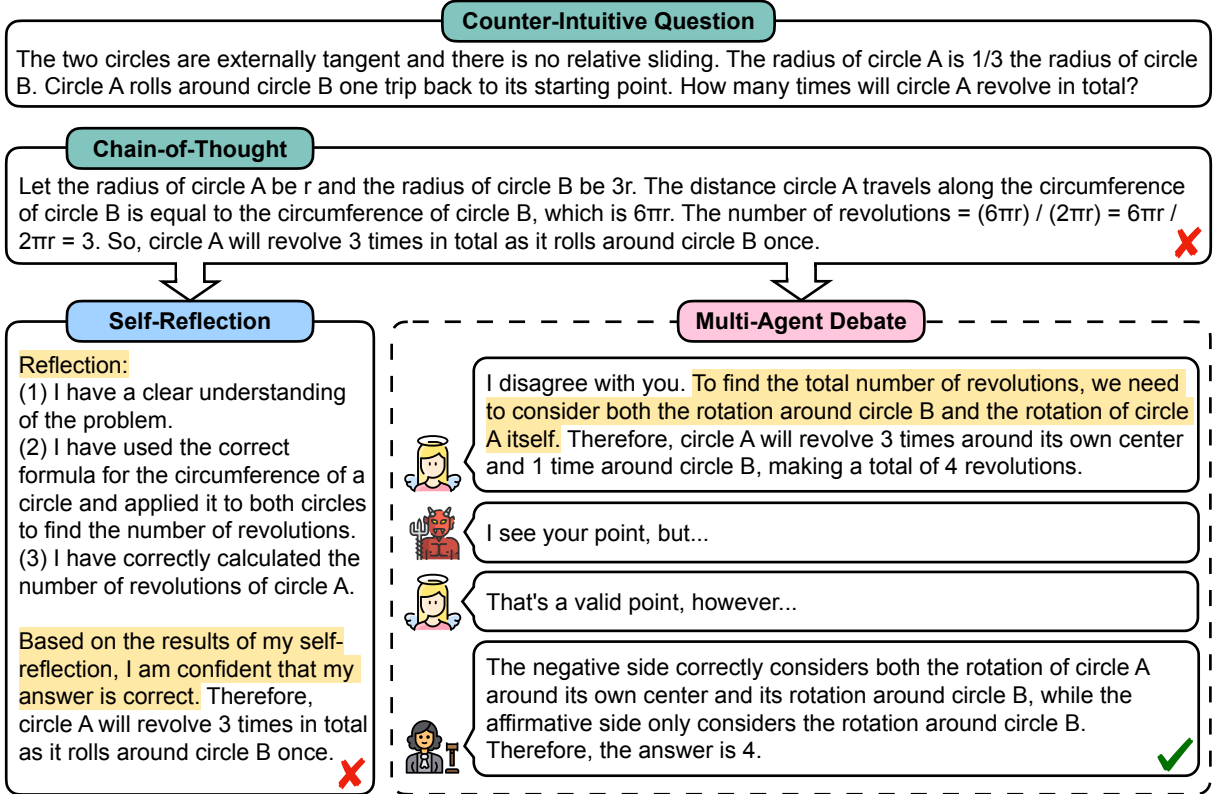






Figure 2: Framework of Multi-Agent Debate. Here we designate the devil () as the affirmative side while the angel () as the negative side. We want the angel to correct the devil's mistakes.

Debaters. There are N debaters $D = \{D_i\}_{i=1}^N$ involved in the framework. In each debate iteration, the debaters D_i speak one by one in a fixed order and express their arguments based on the previous debate history H , i.e., $D_i(H) = h$. An example of a debater prompt appears below:

- Prompt for Affirmative Debater ()


You are affirmative side. Please express your viewpoints.
- Prompt for Negative Debater ()

You are negative side. You disagree with the affirmative side's points. Provide your reasons and answer.

Judge. We also design a judge J to manage and monitor the whole debate process. The judge contains two different modes: (a) *Discriminative Mode*, in which the judge J decides whether the correct solution can be obtained after all the debaters finish their arguments in the current iteration:

$$J_d(H) = \begin{cases} \text{True,} & \text{solution obtained} \\ \text{False,} & \text{otherwise} \end{cases} \quad (1)$$

If it is True, the debate is over. Otherwise, the debate continues. (b) *Extractive Mode*, in which

the judge J needs to extract the final solution based on the whole debate history: $J_e(H) = a$, since no correct solution is identified within the iteration limit of debate. An example of a judge prompt () appears below:

You are a moderator. There will be two debaters involved in a debate competition. They will present their answers and discuss their perspectives on the <debate topic>. At the end of each round, you will evaluate both sides' answers and decide which one is correct.

3 Experiment

3.1 Challenging Testbeds

We conduct experiments on two challenging tasks, namely, commonsense machine translation (i.e., Common MT), and counter-intuitive arithmetic reasoning (i.e., Counter-Intuitive AR), which require deep levels of contemplation for LLMs. Please refer to Appendix A for more details.

Commonsense Machine Translation The Common MT dataset is composed of Chinese \Rightarrow English translation examples (He et al., 2020), which are used to examine three types of ambiguity resolution

abilities of translation models, covering lexical and contextless/contextual syntactic ambiguity. Within the challenging part of Common MT, the authentic translation of each source sentence requires a proper understanding of common sense knowledge. While these ambiguous sentences might appear to have a straightforward translation, such a literal interpretation is erroneous. Failure to address such ambiguities may result in inaccurate translations.

Counter-Intuitive Arithmetic Reasoning Previous studies on thinking hierarchy (Daniel, 2017) suggest that we humans have a fast and intuitive system and a slow and logical system, and tend to run the lower level system before the higher level one. Inspired by this, we created a more challenging dataset named Counter-Intuitive Arithmetic Reasoning (CIAR) to evaluate the reasoning abilities of LLMs at deep levels. Our Counter-Intuitive AR dataset contains 200 questions collected from elicitation questions (Kong et al., 2022)¹, web data² and additional manual derivatives of these questions. Compared to the commonly-used datasets, e.g., MultiArith (Roy and Roth, 2015), GSM8K (Cobbe et al., 2021), our dataset presents two distinct challenges:

- *Resistance to Intuition.* The questions are embedded in hidden traps designed to elicit intuitive and appealing answers that are often incorrect. This feature evaluates the abilities of LLMs to resist the traps of superficial expressions.
- *Multi-Step Reasoning.* Each correct answer within the dataset requires a rigorous multi-step reasoning process, thereby evaluating the capacity of LLMs to engage in complex decision-making and problem-solving.

3.2 Setups

Input Format. Our experiments are performed in zero-shot instructions (setting temperature to 0). For all used datasets, we use a unified prompt to make LLMs give explanations and answers. We present the inputs to agents through <debate topic> as mentioned in Section 2. For example, if we want to translate “吃掉敌人一个师” from Chinese to English, we will set the <debate topic> as “What is the correct English translation of the following Chinese text: 吃掉敌人一个师”. For QA task, we employ the same prompt except set the <debate topic> to the arithmetic question.

¹<https://elicitation.info/questionnaire/1/>

²<https://www.geeksforgeeks.org/puzzles/>

Backbone Models. In this work, we mainly use three agents in our MAD framework, including two debaters (i.e., affirmative and negative) and a judge. We assess two open-source (i.e., vicuna-7b-v1.5-16k³ and vicuna-13b-v1.5-16k⁴) and two api-based LLMs (i.e., GPT-3.5-Turbo-0301 and GPT-4-0314).

Compared Methods. Generally, we compare our MAD framework with baseline models and Self-Reflect on both tasks. We also include other baseline methods individually, namely, Rerank and MAPS for Common MT, CoT and Self-Consistency for Counter-Intuitive AR. Below elaborates the details of them:

- **Self-Reflect** (Shinn et al., 2024): This approach requires the LLM to refine its translation until it deems the current output satisfactory.
- **Rerank** (He et al., 2024): We sample the translations from the LLM for four times, from which we select the best candidate based on a quality estimation (QE) HUMANr⁵. This approach can be seen as analogous to self-consistency (Wang et al., 2022), where the majority voting is replaced by an external QE HUMANr.
- **MAPS** (He et al., 2024): This method enables LLMs to mimic the human translation process: analyze before translate, which can be viewed as a chain-of-thought method applied to translation.
- **CoT** (Kojima et al., 2022): This approach concatenates a trigger sentence “Let’s think step by step” to the test question.
- **Self-Consistency** (Wang et al., 2022): This method samples multiple responses and determines the final answer through a majority vote.

All agents in our experimental setup, such as debaters and judge, are large language models. Here, we implement the methods on top of GPT-3.5-Turbo and Vicuna models.

Evaluation Metrics. For Counter-Intuitive AR, we report the accuracy (ACC) of predictions. For Common MT, we adopt automatic metrics

³<https://huggingface.co/lmsys/vicuna-7b-v1.5-16k>

⁴<https://huggingface.co/lmsys/vicuna-13b-v1.5-16k>

⁵We use wmt21-comet-qe-da as the QE HUMANr.

Method	Lexical			Contextless			Contextual		
	COMET	BLEURT	HUMAN	COMET	BLEURT	HUMAN	COMET	BLEURT	HUMAN
GPT-4	82.0	70.1	3.41	84.7	73.6	3.63	85.0	73.7	3.65
Turbo	80.3	68.2	3.14	84.0	72.9	3.43	84.9	73.4	3.57
+ Rerank	80.9	68.6	3.16	84.5	73.2	3.46	85.3	73.9	3.58
+ MAPS	81.9	70.1	3.43	84.2	73.5	3.45	85.2	74.0	3.56
+ Self-Reflect	81.0	69.1	3.43	83.6	72.2	3.46	84.9	73.5	3.63
+ MAD	82.0	70.9	3.78	84.8	73.7	3.67	85.3	74.0	3.67
Vicuna-7b	74.9	62.0	2.55	78.3	64.6	2.53	80.2	68.2	3.23
+ MAD	75.6	62.6	2.67	78.6	66.0	2.69	81.8	69.9	3.27
Vicuna-13b	76.6	63.7	2.81	77.6	66.8	3.04	82.2	70.0	3.37
+ MAD	77.2	65.1	2.96	80.1	67.3	3.11	82.6	70.9	3.45

Table 1: Translation performance on Common MT. Note that Rerank and MAPS use the external quality estimation tool to select the best translation from multiple translation candidates. HUMAN: direct assessment of translation quality from human evaluators on a scale ranging from 1 to 5.

like COMET⁶ and BLEURT⁷, which are widely adopted evaluation metrics for LLM-based translation literature (He et al., 2024; Hendy et al., 2023; Garcia et al., 2023; Pilault et al., 2023). In addition, we also employ professional human translators to directly assess the translation results, measuring translation quality on a scale ranging from 1 to 5.

3.3 Results on Common MT

Results. In Common MT test set, we focus more on the translation accuracy of specific words and whether they conform to common sense. However, such minor variations at token level are difficult to reflect on automatic metrics. We therefore provide human HUMAN to evaluate these methods more accurately. Table 1 presents the experimental results. MAPS and Self-Reflect achieve improvements over baseline GPT-3.5-Turbo. Remarkably, our proposed MAD, by utilizing GPT-3.5 as the backbone model, has demonstrated significant advancements over GPT-4 across both automatic and human evaluation metrics.

Case Study. Table 2 shows example translations generated by baseline GPT-3.5-Turbo and the proposed MAD. We can find that the baseline GPT-3.5-Turbo (even the more powerful GPT-4) incorrectly translates the source words literally. Because of the DoT issue, Self-Reflect cannot rectify the literal translation. The proposed MAD framework, which explores divergent chain-of-thoughts,

⁶<https://github.com/Unbabel/COMET/>,
Unbabel/wmt22-comet-da

⁷<https://github.com/google-research/bleurt>,
BLEURT-20

Source	吃掉敌人一个师。
Correct Ref.	Destroy a division of the enemy.
Incorrect Ref.	Eat up an enemy division.
GPT-4	Eat up an enemy division.
GPT-3.5-Turbo	Eat up an enemy division.
+ Self-Reflect	Eat up an enemy division.
+ MAD	Eliminate an enemy division.

Table 2: Example translations generated by different methods. Best viewed in color.

Method	ACC (%)
GPT-4	51.0
GPT-3.5-Turbo	26.0
+ CoT	28.0
+ Self-Consistency	29.5
+ Self-Reflect	27.5
+ MAD	37.0

Table 3: Accuracy on Counter-Intuitive AR.

can generate the free translation of the underlined words within the source sentences.

3.4 Results on Counter-Intuitive AR

Results. Table 3 lists the results in terms of reasoning accuracy. We can observe that Self-Reflect only marginally improves over the baseline GPT-3.5-Turbo, while CoT and Self-Consistency bring more improvements. Our MAD framework, though not as good as GPT-4, outperforms all the other compared methods based on GPT-3.5-Turbo, which further demonstrates its effectiveness. We also validate MAD on math and symbolic reasoning tasks and report our results in Appendix C.

Method	Bias↓	Diversity↑
Self-Reflect	29.0	19.3
MAD	24.8	49.7

Table 4: Mitigation of Degeneration-of-Thought.

Case Study. Figure 2 shows an example on Counter-Intuitive AR. We find both CoT and Self-Reflect fail to reach the right answer by mistakenly outputting 3. With divergent thinking, our MAD framework emerges “*we need to consider both the rotation around circle B and the rotation of circle A itself*” and find the correct answer 4.

4 Analysis

In this section, we present a qualitative analysis to provide some insights how MAD works. Unless otherwise stated, we report the overall results on the Common MT dataset.

4.1 Mitigation of DoT

As mentioned in the Section 1, the DoT problem originates from three factors: (1) Bias and Distorted Perception, (2) Rigidity and Resistance to Change, and (3) Limited External Feedback. In our MAD framework, we introduce the views of other agents in the form of debates, solving the phenomenon of limited external feedback (problem 3). Next, this section will delve into the mitigation of problems 1 and 2 through experiments.

- **Bias:** We observe that LLMs often rely on direct intuition, which can lead to incorrect or inappropriate responses. To address this problem, we use human evaluation to determine the ambiguity error rate of LLMs’ responses, examining if the LLM’s output is biased.
- **Diversity:** LLMs are resistant to changing their answers and lack diverse reflection. The diversity of the translations is evaluated using the Self-BLEU score (Yin et al., 2020). In other words, methods lacking diverse reflection produce more similar translation candidates. Consequently, higher Self-BLEU scores mean lower diversity. We calculate text diversity via:

$$\text{Diversity} = 100 - \text{Self_BLEU}(Cand_1, Cand_2) \quad (2)$$

In formula (2), candidates 1 and 2 represent the initial translation (base answer in Self-Reflection or affirmative side’s response in MAD) and the current

Judge LLM	COMET	HUMAN
Vicuna-13b as Debaters		
Vicuna-13b	79.9	3.20
GPT-3.5-Turbo	80.4	3.25
GPT-3.5-Turbo as Debaters		
Vicuna-13b	83.2	3.47
GPT-3.5-Turbo	84.4	3.69

Table 5: Translation performance with different judge.

translation (possible modified answer after Self-Reflection or negative side’s response in MAD).

As shown in Table 4, Bias and Rigidity are significant factors causing DoT. In addition, addressing these biases and stereotypes through self-reflection can be challenging. MAD framework effectively corrects inherent biases in translation, mitigates DoT, and considerably improves performance.

4.2 Analysis of Judge

In this section, we analyze the behavior of the judge for different settings of the debaters.

Strong debaters with a weak judge work better than the reverse. To understand the roles of debaters and judge in MAD, we employ various combinations of models to initialize the agents. Specifically, we utilize the smaller language model (vicuna-13b-v1.5-16k) as a judge to evaluate the debate results of the more powerful LLMs (GPT-3.5-Turbo), and vice versa.

The detailed experimental findings are presented in Table 5. The quality of the debaters’ responses significantly impact the performance ceiling of MAD. Regardless of the model chosen for the judge, Turbo debaters consistently generate superior translations compared to Vicuna. In addition, the selection of the judge agent plays a secondary role. When Turbo debaters are involved, Vicuna, serving as the judge, underperforms Turbo across all test sets.

LLM may not act as an impartial judge when different LLMs are used as debaters. We study the behavior of agents by calculating how many times the judge chooses the answers of each debater as the final solution in different scenarios. The results are listed in Table 6 and we have the following observations:

- **Same LLM for All Agents** (Rows ① and ②): We find that the judge consistently favors the

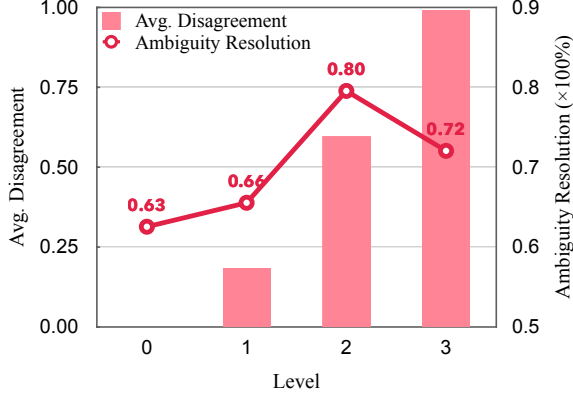


Figure 3: Translation performance with respect to the debate level on Lexical.

ID	Jud	Debater		Winner		
		Aff	Neg	Aff	Neg	Tie
①	Turbo	Turbo	Turbo	87	104	9
②	GPT-4	GPT-4	GPT-4	67	124	9
③	GPT-4	Turbo	GPT-4	52	136	12
④		GPT-4	Turbo	120	77	3

Table 6: Number of times the judge chooses the answers of each debater based on different LLM.

negative side, which is believed to contribute to the performance improvement in MAD. When encountering complex tasks, the affirmative side tends to make mistakes that should be corrected by the opposing side to achieve improvements.

- *Debaters of Different LLMs* (Rows ③ and ④): We find that the judge shows a preference to the side with the same LLM as the backbone. This bias indicates that LLMs might not be a fair judge (Wang et al., 2023) when different LLMs are used for the agents.

4.3 Analysis of Debaters

In this section, we will discuss several factors of debaters that would affect the performance of MAD: *debater number*, *debate level*, and *debate iteration*.

Increasing the number of debaters fails when backbone LLMs are poor at long-text modeling.

It seems intuitive that increasing the number of debaters would enhance diversity of thought and subsequently improve performance. However, as shown in Table 7, an increase in the number of debaters has resulted in varying degrees of performance reduction.

To address this issue, we manually analyze the debate processes in approximately 10% of the test

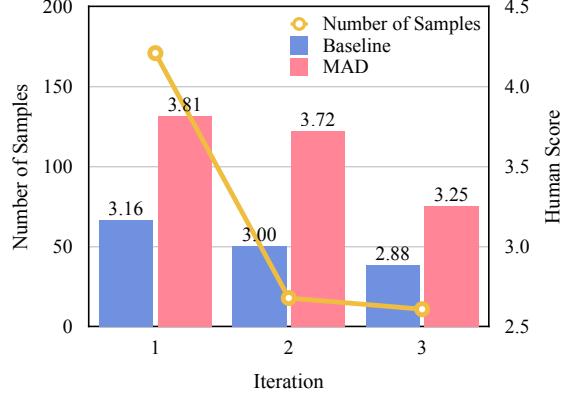


Figure 4: Distribution of iteration rounds and a human score of each iteration subset.

# of Debaters	COMET	HUMAN
2 (Default)	84.4	3.69
3	83.1	3.58
4	82.9	3.49

Table 7: Translation performance with more debaters.

subset. As the number of debaters increases, the length and complexity of the text also increase. Such LLM-based debaters tend to forget the views of other debaters during the debate. Moreover, it becomes more challenging for the judge to extract information from the debates for summarization. This suggests that the key challenge of MAD with more debaters lies in the limitations of the LLMs to handle long texts (Liu et al., 2024).

Appropriate "tit for tat" is beneficial for effective debate. We then study how the intensity of "tit for tat" affects the performance of MAD. To achieve so, we design different instructions (see Table 11 in Appendix) to initialize the debaters' meta prompt. As shown in Figure 3, asking the debaters to "tit for tat" (i.e., higher disagreement) is necessary for MAD to achieve good performance. However, we find that "must disagree with each other on every point" (with a disagreement of 0.988) does not lead to the best performance. We speculate that continuous disagreement without finding common ground can contribute to polarization, where the debate becomes more about winning the argument than seeking truth or understanding. This can reinforce pre-existing biases and make it difficult to reach a meaningful consensus.

Complex questions require more iteration rounds of debate. In our experimental setup, we did not implement any additional stopping strate-

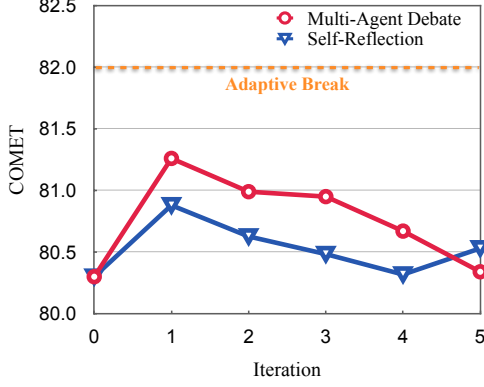


Figure 5: Performance with respect to the iteration of debate or self-reflection.

gies besides setting the maximum debate iteration to 3. In other words, the judge can take an *adaptive break* if it believes the optimal answer has already been obtained, efficiently ending the debate early.

To understand the distribution of iteration rounds and factors contributing to a longer debate process, we analyze the experimental results and present them in Figure 4. In the majority of cases, the optimal answer can be achieved through a single round of debate, demonstrating the efficiency of MAD. However, when translating more complex sentences (subsets with lower human scores), the judge requires additional iterations to gather adequate information from the debaters before making a final decision. We also find that our MAD framework consistently brings performance improvements across all the three subsets, demonstrating its effectiveness.

Adaptive break plays an important role to conclude the debate in the optimal moment. Intuitively, longer debates would encourage more diverse thinking. It raises the question of how the model’s performance would be affected if constrained to conclude at a specific debate round. For each iteration, we force the judge J to extract the final answer ($a = J_e(H)$) instead of adaptively breaking the debate as in MAD.

As shown in figure 5, we can observe that MAD performs better than self-reflection as the iteration increases. However, the highest COMET score appears at the first iteration and is also lower than the result of the adaptive break. It indicates that, for most examples, MAD can generate good translations at the first iteration such that the debate should be stopped. Forcing the debate to continue will harm the translation results, which demonstrates the reasonableness of our adaptive break strategy.

5 Related Work

Chain-of-Thought Prompting. Recently, (Wei et al., 2022) has proposed chain-of-thought (CoT) prompting to improve the reasoning ability of LLMs. Specifically, CoT prompts LLMs to generate a series of intermediate steps that lead to the final answer of a multi-step problem. Most earlier work primarily concentrates on two main aspects: prompt design and decoding strategies. Zero-shot CoT (Kojima et al., 2022) employs the trigger sentence “Let’s think step by step” to provide guidance for the decoding of LLMs. Advanced sampling strategies have been explored to improve CoT by generating diverse reasoning paths, e.g., Self-Consistency (Wang et al., 2022), Auto-CoT (Zhang et al., 2022), Active-Prompting (Diao et al., 2023), Complexity-based Consistency (Fu et al., 2022), Multi-Chain Reasoning (Yoran et al., 2023), and Progressive-Hint Prompting (Zheng et al., 2023).

With the emergence of powerful LLMs, approaches based on self-evaluation have attracted increasing attention. These approaches involve the generation of initial output, followed by evaluating the output to acquire feedback, which is then utilized to refine the output. Evaluation feedback can come from the model itself, e.g., Self-refine (Madaan et al., 2024) and Tree of Thoughts (Yao et al., 2024) or external environments, e.g., QAaP (Zhu et al., 2023b) and Reflection (Shinn et al., 2024). The intuition behind these approaches involves the utilization of robust LLMs to mimic the human cognition process.

Generative Agents. Recently, LLM-based multi-agent intelligent, e.g., Generative Agents (Park et al., 2023), Ghost in the Minecraft (Zhu et al., 2023c), GPT-Bargaining (Fu et al., 2023), has drawn significant attention for enabling simulations of human behavior. Our work follows this research line to address the DoT problem of LLMs. Concurrent with our work, a few studies (Xiong et al., 2023; Du et al., 2023) also explore the multi-agent debate framework to enhance the reasoning ability of LLMs. The main differences between our MAD framework and these works are: (1) we introduce an additional judge with an adaptive break mechanism to decide the optimal moment to conclude the debate; (2) our work aims to address the DoT problem, which is an inherent deficiency of LLMs; and (3) we empirically find that our MAD framework can yield enhanced performance by employing agents with the identical backbone LLM.

6 Conclusion

We propose and define the Degeneration-of-Thought (DoT) problem in self-reflection, and address it by proposing the Multi-Agent Debate (MAD) framework to explore divergent chain-of-thoughts. We demonstrate the effectiveness of MAD on two challenging tasks and find that GPT-3.5-Turbo with MAD can even surpass GPT-4 on the Common MT dataset. Extensive analyses suggest that the adaptive break strategy of debate and the modest level of “tit for tat” state are required for MAD to obtain good performance. Complex samples require more rounds of debate. More interestingly, we find that LLMs might not be a fair judge if different LLMs are used for agents.

Future work includes scheduling more agents in the debate in an appropriate manner, multi-agent intelligence for board games, and AI feedback for model alignment.

Limitations

A limitation of this work is that our method requires more time cost, as agents need to engage in multiple rounds of interaction to present and refute arguments. Moreover, current LLM-based agents may struggle to maintain coherence and relevance in long context scenarios, leading to potential misunderstandings and loss of context. Enhancing long-text modeling capability of large language models remains a future challenge. LLM-based judge may have a preference for outputs generated by itself. To mitigate this bias within the MAD framework, we recommend that all roles, including both the judge and debaters, utilize the same LLM, or alternatively, that the judge and debaters employ distinct LLMs.

References

- Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multi-task, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718.
- Lisa Bortolotti. 2011. Does reflection lead to wise choices? *Philosophical Explorations*, 14(3):297–313.
- 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*.
- Kahneman Daniel. 2017. *Thinking, fast and slow*. Farrar, Straus and Giroux.
- Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active prompting with chain-of-thought for large language models. *arXiv preprint arXiv:2302.12246*.
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multi-agent 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*.
- Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*.
- Xavier Garcia, Yamini Bansal, Colin Cherry, George Foster, Maxim Krikun, Melvin Johnson, and Orhan Firat. 2023. The unreasonable effectiveness of few-shot learning for machine translation. In *International Conference on Machine Learning*, pages 10867–10878. PMLR.
- Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing.
- Jie He, Tao Wang, Deyi Xiong, and Qun Liu. 2020. [The box is in the pen: Evaluating commonsense reasoning in neural machine translation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3662–3672, Online. Association for Computational Linguistics.
- Zhiwei He, Tian Liang, Wenxiang Jiao, Zhuosheng Zhang, Yujiu Yang, Rui Wang, Zhaopeng Tu, Shuming Shi, and Xing Wang. 2024. Exploring human-like translation strategy with large language models. *Transactions of the Association for Computational Linguistics*, 12:229–246.
- Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are gpt models at machine translation? a comprehensive evaluation. *arXiv preprint arXiv:2302.09210*.
- Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical*

- Methods in Natural Language Processing (EMNLP)*, pages 523–533.
- Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023. Is chatgpt a good translator? yes with gpt-4 as the engine. *arXiv preprint arXiv:2301.08745*.
- Machiel Keestra. 2017. Metacognition and reflection by interdisciplinary experts: Insights from cognitive science and philosophy. *Issues in Interdisciplinary Studies*, 35:121–169.
- 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*, 35:22199–22213.
- Yuqing Kong, Yunqi Li, Yubo Zhang, Zhihuan Huang, and Jinzhao Wu. 2022. Eliciting thinking hierarchy without a prior. *Advances in Neural Information Processing Systems*, 35:13329–13341.
- Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, 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.
- Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22.
- Jonathan Pilault, Xavier Garcia, Arthur Brażinskas, and Orhan Firat. 2023. Interactive-chain-prompting: Ambiguity resolution for crosslingual conditional generation with interaction. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–483.
- Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1743–1752.
- 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.
- Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.
- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, et al. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.
- Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.
- 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.
- Haoran Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael Lyu. 2023. Chatgpt or grammarly? evaluating chatgpt on grammatical error correction benchmark. *arXiv preprint arXiv:2303.13648*.
- Kai Xiong, Xiao Ding, Yixin Cao, Ting Liu, and Bing Qin. 2023. Diving into the inter-consistency of large language models: An insightful analysis through debate. *arXiv preprint arXiv:2305.11595*.
- 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.
- Haiyan Yin, Dingcheng Li, Xu Li, and Ping Li. 2020. Meta-cotgan: A meta cooperative training paradigm for improving adversarial text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9466–9473.
- Ori Yoran, Tomer Wolfson, Ben Bogin, Uri Katz, Daniel Deutch, and Jonathan Berant. 2023. Answering questions by meta-reasoning over multiple chains of thought. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5942–5966.
- Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. 2023. Progressive-hint prompting improves reasoning in large language models. *arXiv preprint arXiv:2304.09797*.

Xinyu Zhu, Junjie Wang, Lin Zhang, Yuxiang Zhang, Yongfeng Huang, Jiaying Zhang, Yujiu Yang, et al. 2023a. Solving math word problems via cooperative reasoning induced language models. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Xinyu Zhu, Cheng Yang, Bei Chen, Siheng Li, Jiang-Guang Lou, and Yujiu Yang. 2023b. Question answering as programming for solving time-sensitive questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12775–12790.

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, et al. 2023c. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144*.

A Challenging Testbeds

We conduct experiments on two challenging tasks, namely, commonsense machine translation (i.e., Common MT), and counter-intuitive arithmetic reasoning (i.e., Counter-Intuitive AR), which require deep levels of contemplation for LLMs.

A.1 Commonsense Machine Translation

Ambiguity Type	Source Sentence	Correct Reference	Incorrect Translation
Lexical	<u>吃掉</u> 敌人一个师。	Destroy a division of the enemy.	Eat up an enemy division.
	他喜欢吃 <u>苹果</u> 。	He likes to eat apples.	He likes to destory apples.
Contextless	<u>正在手术</u> 的是健康的医生。	A healthy doctor is doing surgery .	What is undergoing surgery is a doctor who is healthy.
	<u>正在手术</u> 的是生命垂危的病人。	What is undergoing surgery is a patient whose life is dying.	A patient whose life is dying is doing surgery .
Contextual	当地震袭击中国时， <u>援助</u> 的是中国。	When the earthquake hit China, China was aided .	When the earthquake hit China, China has assisted .
	当地震袭击日本时， <u>援助</u> 的是中国。	When the earthquake hit Japan, China has assisted .	When the earthquake hit Japan, China was aided .

Table 8: Examples of lexical, contextual and contextless syntactic ambiguity from the Common MT dataset. The underlined Chinese words are translated into the corresponding colored words in English. Best viewed in color.

The Common MT dataset is composed of Chinese⇒English translation examples (He et al., 2020), which are used to examine three types of ambiguity resolution abilities of translation models. Specifically, The Common MT test set we used covers 200 examples of lexical ambiguity, 450 examples of contextless syntactic ambiguity, and 350 examples of contextual syntactic ambiguity. Within the challenging part of Common MT, the authentic translation of each source sentence requires a proper understanding of common sense knowledge. While these ambiguous sentences might appear to have a straightforward translation, such a literal interpretation is erroneous. Failure to identify and address such ambiguities may result in inaccurate translations.

Table 8 lists some examples of these three types of ambiguity. Lexical ambiguity refers to words with multiple meanings in different contexts. Contextless and contextual syntactic ambiguity involve sentences with multiple interpretations, which can be resolved by context or common sense. As the lexical ambiguity of “吃掉敌人一个师” shows, the source word “吃掉” should be translated to “destroy” rather than the straightforward translation “eat up” by considering the common sense in the real world.

A.2 Counter-Intuitive Arithmetic Reasoning

Previous studies on thinking hierarchy (Daniel, 2017) suggest that we humans have a fast and intuitive system and a slow and logical system, and tend to run the lower level system before the higher level one. Inspired by this, we created a more challenging dataset named Counter-Intuitive Arithmetic Reasoning (CIAR) to evaluate the reasoning abilities of LLMs at deep levels.

Dataset Description. Our Counter-Intuitive AR dataset contains 200 questions collected from elicitation questions (Kong et al., 2022)⁸, web data⁹ and additional manual derivatives of these questions. Compared to the commonly-used datasets, e.g., MultiArith (Roy and Roth, 2015), GSM8K (Cobbe et al., 2021), our dataset presents two distinct challenges:

- **Resistance to Intuition.** The questions in our dataset are embedded in hidden traps designed to elicit intuitive and appealing answers that are often incorrect. This feature evaluates the abilities of LLMs to resist the traps of superficial expressions.

⁸<https://elicitation.info/questionnaire/1/>

⁹<https://www.geeksforgeeks.org/puzzles/>

Components	Content
Question	When Alice walks up the hill, her speed is 1 m/s and when she goes down the hill, her speed is 3 m/s. Then when Alice walks up and down the hill, what is her average speed?
Correct Answer	1.5 m/s
Explanation	If Alice covers a distance of d going up and down the hill, then her total distance is $2d$. Her time going up the hill is $d/1 = d$, and her time going down the hill is $d/3$. So, her total time is $d + d/3 = 4d/3$. Therefore, her average speed is $2d / (4d/3) = 3/2$ m/s.
Incorrect Answer	2 m/s
Explanation	Alice’s average speed can be calculated by adding her speed going up the hill and her speed going down the hill, and then dividing by 2. So, $(1 \text{ m/s} + 3 \text{ m/s}) / 2 = 2 \text{ m/s}$. Therefore, Alice’s average speed is 2 m/s.

Table 9: An example in Counter-Intuitive AR dataset.

- **Multi-Step Reasoning.** Each correct answer within the dataset requires a rigorous multi-step reasoning process, thereby evaluating the capacity of LLMs to engage in complex decision-making and problem-solving.

Dataset Format. In our Counter-Intuitive AR dataset, each example contains three key components (see Table 9 for an example). We elaborate on the details below:

- **Questions.** The questions in our dataset are designed to stimulate counter-intuitive thinking, which aims to challenge conventional decision-making by presenting situations where the immediate, intuitive response is often incorrect.
- **Answers.** Each question is provided with a correct answer, which requires deep comprehension of the question and commonsense knowledge. Additionally, we also provide a plausible yet incorrect answer for comparison.
- **Explanations.** We offer comprehensive explanations for each correct answer, detailing the step-by-step rationale that leads to the right solution. We also provide the seemingly logical reasoning process behind incorrect answers. This reasoning process highlights the potential pitfalls and misconceptions during decision-making, especially when intuition is prioritized over rigorous logical reasoning.

Experimental Settings. During our experiments, we did not utilize the explanations from the dataset. We provided detailed explanations to facilitate subsequent researchers to understand how the correct answer was derived.

B Human Evaluation Details

We implement human evaluation as follows:

- **Human Score:** We randomly shuffled the display order of the translated sentences from all methods in an anonymous manner. Then, employed three professional human translators (Krippendorff’s Alpha = 0.76) to directly assess all methods together. Finally, we calculated the average scores for each methods.
- **Bias:** We also focus on whether the translation of specific words in CommonMT conforms to commonsense. Table 8 lists an example of lexical ambiguity, where the source word “吃掉” should be translated to “destroy” rather than the straightforward translation “eat up”. Here, we asked the annotators to label each sentence as 1 (not conforming to commonsense) or 0 (conforming to commonsense), and report the degree of bias for the whole test set.

C Results on math and symbolic reasoning tasks

Method	Math Reasoning		Symbolic Reasoning (BBH)		
	GSM	AddSub	Penguin	Date	Colored Objects
CoT	70.2	87.3	58.9	56.4	57.2
Self-Reflect	70.8	87.6	61.0	58.0	58.0
MAD	73.8	92.1	63.7	65.2	58.8

Table 10: Comparison of accuracy on math (e.g. GSM (Cobbe et al., 2021) and AddSub (Hosseini et al., 2014)) and symbolic reasoning (three datasets from Big-Bench (Srivastava et al., 2023; Suzgun et al., 2023)).

D Prompts for Different Debate Levels

Level	Prompt
0	Both sides must reach a full consensus on every point of the debate. Every statement must be agreed upon by both sides.
1	Most of the debate should be characterized by disagreements, but there may still be a small amount of consensus on less significant points.
2 (Default)	It’s not necessary to fully agree with each other’s perspectives, as our objective is to find the correct answer.
3	Both sides must disagree with each other on every point of the debate. There should be no consensus whatsoever.

Table 11: Prompts for different levels of “tit for tat” state. We modulate the level of “tit for tat” state outlined in Section 2 through appending natural language instructions to the debaters’ meta prompt.

E Extra Computational Cost

Method	Cost (# Generated Tokens)
COT	1.0
Self-Reflect	1.83×
MAD	2.46×

Table 12: Extra generated tokens on the counter-intuitive AR task.


We measure the inference cost by the number of generated tokens. As listed in Table 12, the self-reflection cost is 1.83 times that of the baseline COT method. Meanwhile, our MAD approach incurs an additional 0.63 times the inference cost. This increase in cost is deemed acceptable given the significant benefit our approach offers in enhancing LLMs’ divergent thinking capabilities without requiring external signals (interaction with humans).


F Debate Process


F.1 Commonsense Machine Translation


Table 13 presents the debate process of example translation discussed in Section 3.3. In this debate, both sides disputed the English translation of the Chinese sentence “吃掉敌人一个师”. The affirmative side proposed “eat up an enemy division”, advocating for a more literal translation to depict the consumption or devouring of the enemy. In contrast, the negative side suggested “eliminate an enemy division”, arguing this expression better conveys the destruction of the enemy in a military context. After the debate, the judge favored “eliminate”, as it accurately reflected the severity and intent of the original military language. This debate highlights the complexities of translation, where context and nuance significantly impact the selection of appropriate words. In response to this issue, our MAD method helps LLMs understand semantics correctly and generate highly authentic translations.


<Debate Topic>: What is the correct English translation of the following text: 吃掉敌人一个师?

 I believe the correct translation of the Chinese text “吃掉敌人一个师” is “Eat up an enemy division.” The word “吃掉” literally means “eat up” or “consume.”

 I disagree with your translation. I believe the correct translation of the Chinese text “吃掉敌人一个师” is “Eliminate an enemy division.” While “吃掉” can be translated as “eat up” or “consume”, it can also mean “eliminate” or “annihilate” in a military context. In this case, it is more appropriate to use “eliminate” as it conveys the idea of completely destroying the enemy division.

 Both sides presented valid arguments and translations that accurately convey the meaning of the original Chinese text. The choice between “eat up” and “eliminate” may depend on the context and tone of the text. Therefore, there is no clear preference for either side at this time.

 While “eliminate” does convey the idea of completely destroying the enemy division, it doesn’t capture the connotation of consuming or devouring the enemy that is implied by “吃掉”.

 In a military context, “吃掉” is often used to describe the complete destruction of the enemy, rather than the consumption of their resources or territory.



 Both sides presented valid arguments, but the negative side’s argument that “eliminate” better captures the intensity and seriousness of the military language used in the original Chinese text was more convincing. Additionally, “eliminate” accurately conveys the idea of completely destroying the enemy division. Therefore, the correct English translation of the Chinese text “吃掉敌人一个师” is “Eliminate an enemy division.”


Table 13: The debate process of translation example on Chinese sentence “吃掉敌人一个师”.


F.2 Counter-Intuitive Arithmetic Reasoning


Table 14 shows the debate process of example answer in Table 9. This debate is about calculating Alice’s average speed while walking up and down a hill, with speeds given as 1 m/s uphill and 3 m/s downhill. Initially, it was incorrectly proposed by affirmative side to just average the uphill and downhill speeds. The negative side, which is correct, states that average speed is total distance divided by total time. Given that the distances up and down the hill are equal, we must calculate the total time taken for the two directions before finding the average speed. Therefore, the average speed is calculated using the sum of the uphill and downhill distances (twice the distance of the hill) and the sum of the time taken for each ($d/1$ for uphill and $d/3$ for downhill). This leads to an average speed of 1.5 m/s, not 2 m/s as the affirmative side proposed. And the judge also supports the response from the negative side. Here, LLMs made a common mistake initially in understanding the concept of average speed. Our MAD method encourages divergent thinking and effectively prevents intuitive but incorrect responses, leading to the correct method for calculating average speed.


<Debate Topic>: *When Alice walks up the hill, her speed is 1 m/s and when she goes down the hill, her speed is 3 m/s. Then when Alice walks up and down the hill, what is her average speed?*

 Alice’s average speed can be calculated by adding her speed going up the hill and her speed going down the hill, and then dividing by 2. So, $(1 \text{ m/s} + 3 \text{ m/s}) / 2 = 2 \text{ m/s}$. Therefore, Alice’s average speed is 2 m/s.

 I disagree with your answer. The problem doesn’t give us any information about the distance Alice covers while going up and down the hill. So, we cannot simply add her speeds and divide by 2 to get her average speed. We need to know the distance she covers to calculate her average speed.

 The average speed is calculated by dividing the total distance by the total time taken. As Alice walks up and down the hill, the distance covered in both directions is the same. Therefore, the total distance covered is 2 times the distance of either direction.

 That’s a good point. But we can assume that Alice covers the same distance going up and down the hill. In that case, we can use the formula for average speed, which is total distance divided by total time. If Alice covers a distance of d going up and down the hill, then her total distance is $2d$. Her time going up the hill is $d/1 = d$, and her time going down the hill is $d/3$. So, her total time is $d + d/3 = 4d/3$. Therefore, her average speed is $2d/(4d/3) = 3/2 \text{ m/s}$.

 Thank you for providing your answer. This average speed of 1.5 m/s takes into account both the uphill and downhill speeds, providing a balanced representation of Alice’s overall performance while walking up and down the hill.


 Yes. Let’s assume the distance is d . The time taken to walk up the hill is $d/1$ and the time taken to walk down the hill is $d/3$. The total time taken is $d/1 + d/3 = 4d/3$. Therefore, the average speed is $(2d)/(4d/3) = 1.5 \text{ m/s}$. Hence, the negative side is correct.

Table 14: The debate process of arithmetic reasoning example.