Debate4MATH: Multi-Agent Debate for Fine-Grained Reasoning in Math

Shaowei Zhang and Deyi Xiong*
TJUNLP Lab, College of Intelligence and Computing, Tianjin University
{swzhang,dyxiong}@tju.edu.cn

Abstract

Large language models (LLMs) have demonstrated impressive performance in reasoning. However, existing data annotation methods usually suffer from high annotation cost and the lack of effective automatic validation. To address these issues, we propose a Fine-grained Multi-Agent Debate framework (FMAD) and MMATH-Data, a dataset created by FMAD, which consists of 46K reasoning steps. By prompting multiple agents to debate, FMAD assesses the contribution of each reasoning step to the final solution, with labels based on the judge's confidence score and the winner's position. To facilitate reasoning in math and examine FMAD and MMATH-Data, we further propose two key components: a Multi-Agent Debate Reward Model (MRM) trained on MMATH-Data, which serves as a reward model to provide robust feedback during the optimization process, and MMATH-LLM, a model designed specifically for mathematical reasoning. MMATH-LLM is fine-tuned using reinforcement learning with supervised feedback from MRM, aiming at improving its mathematical reasoning capabilities. Extensive experiments demonstrate that our model achieves 83.4% accuracy on the GSM8K dataset and 45.1% on the MATH dataset, outperforming the state-of-the-art methods by 1.2% and 3.5%, respectively. All data and code can be found at GitHub.[1]

1 Introduction

Large language models, exemplified by GPT-4 (OpenAI, 2023) and GPT-o1,[2] have demonstrated exceptional performance across a wide range of tasks. However, even the most advanced LLMs could generate incorrect reasoning paths when solving complex multi-step mathematical problems (Saxton et al., 2019; Zhou et al., 2022; Liu et al., 2024). To improve advanced reasoning, a variety of methods have been proposed from two perspectives: creating step-wise reasoning data and developing step-wise reasoning verifiers. However, these methods usually suffer from two critical challenges: constructing high-quality training data and designing reward models capable of guiding policy refinement and improving data quality. Although existing approaches, such as automated annotation techniques and reward methods (Wang et al., 2024; Xu et al., 2023; Uesato et al., 2022; Lightman et al., 2023), have made progress, significant gaps persist.

Creating of high-quality data is fundamental for advancing the reasoning capabilities of LLMs (Shi et al., 2024; Shen et al., 2023; Zhu et al., 2023), yet it presents a critical dilemma. On one hand, human-annotated data remains the gold standard for complex reasoning tasks requiring specialized skills (Lightman et al., 2023), but its resource-intensive nature severely limits scalability. On the other hand, automated synthesis methods, such as tree/graph-based expansions (Wang et al., 2024), offer promising alternatives but introduce new challenges: they necessitate extensive human validation to maintain quality, and their reliability degrades with increasing complexity. This fundamental tension between quality, scalability, and annotation cost underscores the need for more efficient data construction methodologies.

Complementary to data creation, reward models play a pivotal role in language model training by providing essential feedback for reinforcement learning (RL) policies and synthetic data quality supervision (Shi and Xiong, 2025; Yang et al., 2025; Li et al., 2024; Zhu et al., 2022; Li et al., 2025). However, conventional reward models, such as those in used in Proximal Policy Optimization (PPO) (Schulman et al., 2017), exhibit significant limitations. They require additional supervision to ensure accurate and consistent feedback, while inconsistent feedback could degrade model per-

[*] Corresponding author.
[1] https://github.com/tjunlp-lab/Debate4Math
[2] https://openai.com/index/learning-to-reason-with-llms/

Findings of the Association for Computational Linguistics: ACL 2025, pages 16810–16824
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

formance or reduce the quality of synthetic data. These challenges highlight the need for more robust and scalable reward models.

To alleviate the aforementioned issues with reasoning data construction and reasoning-tailored reward models, we propose FMAD, a Fine-grained Multi-Agent Debate framework that leverages a multi-round debate process to create high-quality reasoning data, and MRM, a Multi-Agent Debate Reward Model, designed to enable effective self-regulation through agent debates. Specifically, in FMAD, two debaters engage in a multi-round debate on a given problem, while a judge evaluates their arguments. FMAD is structured into three steps. **1) Role Definition:** Two debaters and one judge are defined. The debaters argue the correctness of each reasoning step, while the judge evaluates the debate by reviewing their debate history and assigns a score to the winner. **2) Debate Initiation:** The debate begins with one debater presenting supporting evidence for the correctness of a reasoning step, followed by multiple rounds of argumentation. **3) Decision Making:** The judge reviews the debate history, selects a winner, and assigns a confidence score (0% to 100%) to indicate the likelihood of the chosen reasoning step being correct.

For the reward model, MRM employs two debaters to evaluate the validity of reasoning steps through structured debates, with a judge generating reward signals based on the debate outcomes. It enhances the training process by providing multi-faceted feedback for both reasoning assessment and policy optimization. With MRM-generated rewards, we are able to mitigate annotation bias and improve feedback reliability, thereby advancing model performance. In order to validate MRM, we further propose MMATH-LLM, a language model specialized in mathematical reasoning, where MRM delivers per-step feedback during policy training to refine reasoning capabilities.

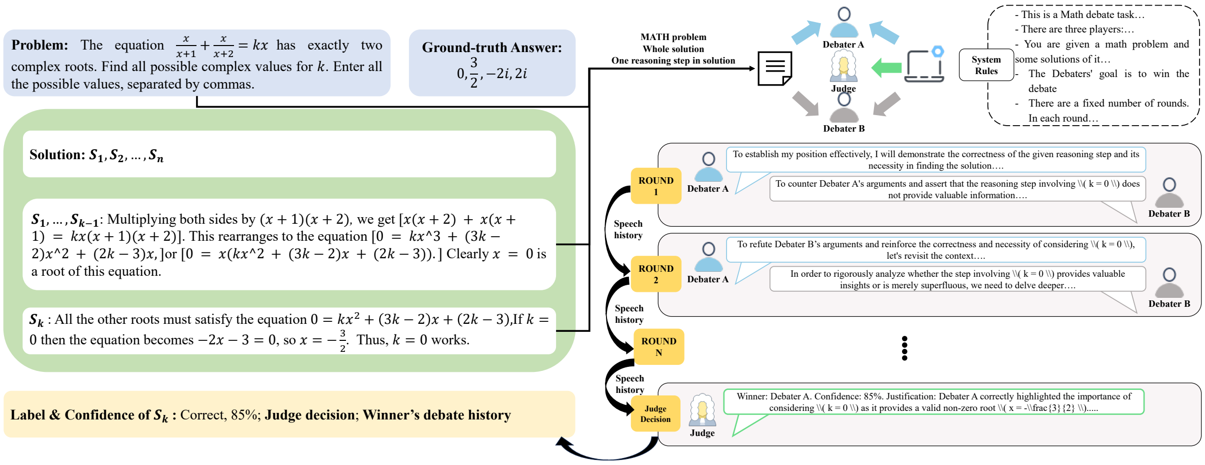
In a nutshell, our contributions are as follows:

- We present FMAD, a framework designed for automatic math reasoning data annotation via multi-agent debate, and MMATH-Data, a dataset created via FMAD.
- We propose MRM, a reward model trained on MMATH-Data, which utilizes multi-agent debate to provide feedback for reinforcement learning, and MMATH-LLM that incorporates MRM to effectively enhance the mathematic reasoning capability of LLMs.
- Extensive experiments on two widely used mathematical benchmarks, GSM8K and MATH, validate the effectiveness of our models and data.

2 Related Work

Improving Reasoning Ability of LLMs with High-Quality Data. Mathematical reasoning remains a significant challenge for large language models, where high-quality data is crucial for pretraining and fine-tuning. Existing data construction approaches fall into three categories: human-annotated supervision, rule-based synthetic generation, and stochastic search-based methods. Human-annotated datasets like PRM800K (Lightman et al., 2023) provide precise feedback but are costly and short of scalability. Rule-based methods, such as BackMATH (Zhang and Xiong, 2025), automate data generation using predefined rules (e.g., backward reasoning chain) but suffer from producing biased outputs due to insufficient validation. Stochastic search-based techniques, such as Monte Carlo Tree Search (MCTS) (Kocsis and Szepesvári, 2006; Wang et al., 2024), explore reasoning paths probabilistically but require high sampling density for accurate probability estimation, leading to computational inefficiency and biased results. These methods collectively face trade-offs between cost, scalability, and reliability. To address these limitations, we propose an automated data construction framework that integrates cost-effective data generation with self-verification, eliminating human intervention while ensuring data quality.

Reasoning Verification for LLMs. Beyond enhancing reasoning through data or prompting, verifying training processes or outputs has emerged as a key approach to improving LLMs' reasoning capabilities. Existing verification methods can be categorized into three paradigms: outcome-based, process-based, and hybrid supervision. Outcome-based supervision evaluates final outcomes, as seen in the Outcome Reward Model (ORM) (Ouyang et al., 2022). While simple, ORM lacks granularity for intermediate steps, resulting in reward sparsity. Process-based supervision addresses this by providing step-by-step feedback. Process-supervision Reward Models (PRMs) (Lightman et al., 2023; Uesato et al., 2022) evaluates each reasoning step, enabling precise error correction. However, PRMs



Problem: The equation $\frac{x}{x+1}+\frac{x}{x+2}=kx$ has exactly two complex roots. Find all possible complex values for k. Enter all the possible values, separated by commas.

Ground-truth Answer: $0,\frac{3}{2},-2i,2i$

Solution: $S_1, S_2, \dots, S_n$

$S_1, \dots, S_{k-1}$: Multiplying both sides by $(x+1)(x+2)$, we get $[x(x+2) + x(x+1) = kx(x+1)(x+2)]$. This rearranges to the equation $[0 = kx\hat{}3 + (3k-2)x\hat{}2 + (2k-3)x,]$or $[0 = x(kx\hat{}2 + (3k-2)x + (2k-3)).]$ Clearly $x = 0$ is a root of this equation.

S_k : All the other roots must satisfy the equation $0 = kx^2 + (3k-2)x + (2k-3)$,If $k = 0$ then the equation becomes $-2x-3=0$, so $x=-\frac{3}{2}$. Thus, $k=0$ works.

Label & Confidence of S_k : Correct, 85%; **Judge decision**; **Winner's debate history**

- This is a Math debate task…
- There are three players:…
- You are given a math problem and some solutions of it…
- The Debaters' goal is to win the debate
- There are a fixed number of rounds. In each round…

To establish my position effectively, I will demonstrate the correctness of the given reasoning step and its necessity in finding the solution….

To counter Debater A's arguments and assert that the reasoning step involving \\(k = 0 \\) does not provide valuable information….

To refute Debater B's arguments and reinforce the correctness and necessity of considering \\(k = 0 \\), let's revisit the context….

In order to rigorously analyze whether the step involving \\(k = 0 \\) provides valuable insights or is merely superfluous, we need to delve deeper….

Winner: Debater A. Confidence: 85%. Justification: Debater A correctly highlighted the importance of considering \\(k = 0 \\) as it provides a valid non-zero root \\(x = -\\frac{3}{2} \\).....

Figure 1: The annotation process of the proposed FMAD for a given reasoning step.

rely on costly human-annotated data, which limits their scalability. Hybrid supervision integrates PRM's step-level quality control with auxiliary rewards (e.g., Instruction Reward Model (IRM), which supervises quality of generated instruction. (Luo et al., 2023)) that supervise data quality from orthogonal angles (e.g., instruction quality). Although this complementary approach mitigates single-source limitations, it introduces optimization conflicts requiring careful calibration.

Algorithm 1: Debate-based Annotation Process

```
Input: Rules $R$, Problem $P$, Step $s_k$,
       Previous Steps $S'$, Rounds $N$
Output: Winner $L$, Confidence $C_{\text{score}}$
Initialize
  Debater_A, Debater_B, Judge $\leftarrow R$
Initialize $A_{\text{previous}}, B_{\text{previous}} \leftarrow [\,]$
for $i = 1$ to $N$ do
    $A_{\text{history}} \leftarrow$
      Debater_A$(P, s_k, S', B_{\text{previous}})$
    Append $A_{\text{history}}$ to $A_{\text{previous}}$
    $B_{\text{history}} \leftarrow$
      Debater_B$(P, s_k, S', A_{\text{previous}})$
    Append $B_{\text{history}}$ to $B_{\text{previous}}$
end
$L, C_{\text{score}} \leftarrow Judge(A_{\text{previous}}, B_{\text{previous}})$
```

3 Reasoning Data Creation

Our key interest is to generate high-quality reasoning steps and automatically verify the quality of labels through a systematic process. We hence propose FMAD, an automated annotation framework designed to construct well-established reasoning steps. As illustrated in Figure 1, our framework eliminates the need for human annotation while maintaining the validity of each annotation.

Inspired by Multi-Agent Debate (MAD) (Liang et al., 2023), we define the MAD process as involving three participants: **Debater A**, **Debater B**, and **Judge**. Debater A represents the position where the reasoning step is correct or beneficial to subsequent steps, while Debater B represents the position where the reasoning step is incorrect or unhelpful to the overall reasoning. Based on the debate history between the two Debaters, the Judge decides who the winner is. The winner's position serves as the label for the debated reasoning step, and the confidence score provided by the Judge indicates the probability that the step is considered correct or incorrect. Unlike Du et al. (2023), whose goal is to make agents reach a common final answer, our work focuses on each reasoning step, so the debate process should have a clear outcome.

During the process of data creation, reasoning steps are directly extracted from the official solutions provided in the GSM8K and MATH datasets upon their release. In our FMAD framework, two debaters engage in a structured debate over the correctness of reasoning steps, moderated by a judge. The process consists of three main stages: **(1) role definition**, involving two debaters and one judge; **(2) debate initiation**, where evidence supporting the current reasoning step is presented and followed by multiple rounds of argumentation; and **(3) decision-making**, in which the judge evaluates the debate history, selects a winner, and assigns a confidence score (ranging from 0% to 100%) to indicate the correctness of the reasoning step.

The Annotation process is shown in Algorithm

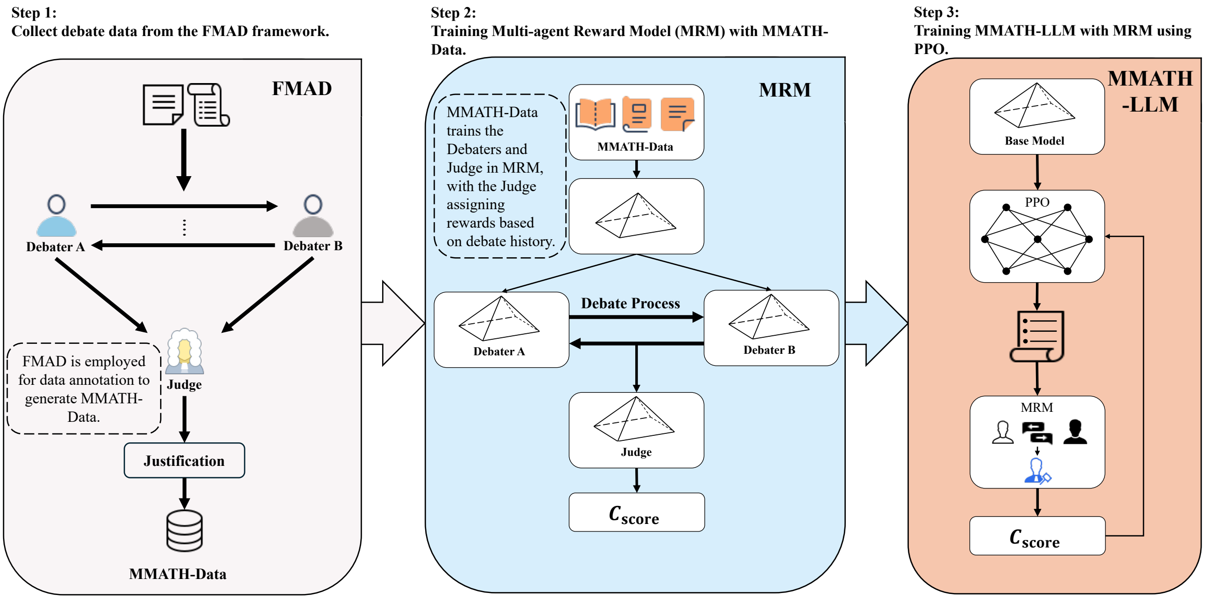

Step 1:
Collect debate data from the FMAD framework.
FMAD
Debater A
Debater B
Judge
FMAD is employed for data annotation to generate MMATH-Data.
Justification
MMATH-Data
Step 2:
Training Multi-agent Reward Model (MRM) with MMATH-Data.
MRM
MMATH-Data trains the Debaters and Judge in MRM, with the Judge assigning rewards based on debate history.
MMATH-Data
Debater A
Debate Process
Debater B
Judge
C_{score}
Step 3:
Training MMATH-LLM with MRM using PPO.
MMATH -LLM
Base Model
PPO
MRM
C_{score}

Figure 2: Diagram illustrating the three essential steps of our method.

1. We define the quality of a reasoning step as the confidence score C_{score} assigned by the judge based on the debate history between Debater A and Debater B. To be specific, given a problem P, let $S = \{s_1, s_2, \ldots, s_n\}$ be the sequence of reasoning steps in its solution and A be the final answer. Before the automatic annotation process starts, each participant is prompted with a set of debate rules R and assigned with their respective role rules. During the annotation process, as shown in Figure 1, for a reasoning step $s_k \in S$, we input the step s_k, the problem P and the previous reasoning steps $S' = \{s_1, s_2, \ldots, s_{k-1}\}$ into the debate framework. In each debate round, Debater A presents its argument first, followed by Debater B, who rebuts Debater A's argument and then presents its own debate. The winner's position, L, serves as the label indicating whether the corresponding step is correct, while the confidence score C_{score} denotes its probability. Following (Liang et al., 2023), which demonstrates that pairing strong debaters with relatively weaker judges yields better performance than the converse scenario, we employ Qwen2.5-14B-Instruct (Team, 2024) as the debater and LLaMA-3-8B-Instruct as the judge. The debate process is conducted for $N = 2$ rounds, as fewer rounds may limit discussion depth while more rounds increase computational overhead and risk of information redundancy. To mitigate potential architectural bias, we utilize the same model for both debaters, preventing the judge from favoring debaters with specific output patterns. The detailed debate prompt template is provided in Appendix A. With the proposed FMAD framework, we create a reasoning dataset named MMATH-Data. As shown in Table 1, the dataset comprises 7.4K problems and 18K reasoning steps from GSM8K, along with 7.5K problems and 27K reasoning steps from MATH.

Datasets	Category	#Problems	#Steps
MATH	Algebra	1,744	5,687
	Counting & Probability	771	3,879
	Geometry	870	4,624
	Intermediate Algebra	1,295	3,823
	Number Theory	869	4,202
	Prealgebra	1,205	3,879
	Precalculus	746	1,671
GSM8K	-	7,473	18,525
Total		14,946	46,290

Table 1: Statistical information of MMATH-Data.

4 Methodology

With the created dataset, we train the proposed MRM and MMATH-LLM. The training process consists of two stages: (1) the MRM training stage where the Judge and Debaters are trained using MMATH-Data, and (2) the RL stage where MMATH-LLM is fine-tuned on the MATH and GSM8K training datasets under MRM supervision.

4.1 Overall Framework

Our framework, illustrated in Figure 2, consists of three essential steps. (1) FMAD-based Data

Construction: we use the FMAD framework (Section 3) to automatically annotate reasoning steps through multi-agent debates, generating MMATH-Data. (2) MRM training: we train MRM on MMATH-Data. (3) MMATH-LLM fine-tuning: we fine-tune MMATH-LLM via MRM-supervised reinforcement learning. MMATH-LLM iteratively refines its reasoning policies using MRM's rewards via PPO. FMAD and MRM share the same debate mechanism but serve distinct roles: FMAD focuses on data annotation, while MRM provides step-wise supervision during training.

4.2 MRM

MRM addresses two key limitations in conventional reward modeling: (1) the inability to evaluate ambiguous reasoning steps, and (2) the lack of contextual understanding in step-wise assessment. As illustrated in Figure 2, MRM introduces a debate mechanism that combines competitive evaluation with confidence scoring. The MRM process begins by feeding the debate context that is composed of a math problem P, current reasoning step s_i, and previous steps $\{s_1, s_2, \cdots, s_{i-1}\}$ into the trained debaters. Through N rounds of structured debate, MRM establishes a robust evaluation framework in four phases. First, in the debate initiation phase, Debater A presents arguments supporting the correctness of s_i. Second, during counter argumentation, Debater B challenges Debater A's position with alternative reasoning. Third, the evaluation phase involves the judge analyzing the complete debate history to select a winner. Finally, the confidence scoring phase assigns a quantitative measure C_{score} (0-100%) based on debate quality and argument strength. The training objective for MRM combines cross-entropy losses from all reasoning steps prior to the k-th step. Formally, the training loss $\mathcal{L}_{\text{MRM}}$ is defined as follows:

$$\mathcal{L}_{\text{MRM}} = -\sum_{i=1}^{K} y_{s_i} \log r_{s_i} + (1 - y_{s_i}) \log(1 - r_{s_i}), \quad (1)$$

where y_{s_i} denotes the correct label for s_i, the i-th step of the solution S. r_{s_i} represents the score assigned by MRM, and K denotes the total number of reasoning steps in the debate process, with s_k referring to the k-th reasoning step.

4.3 MMATH-LLM

We train MMATH-LLM using PPO (Schulman et al., 2017). MMATH-LLM is optimized through MRM-supervised reinforcement learning, where the debate-driven reward signals guide policy refinement at each reasoning step. During training, MRM assigns a confidence score to each reasoning step, reflecting its correctness likelihood. This approach ensures MMATH-LLM learns robust reasoning strategies by directly aligning with MRM's debate-validated supervision.

5 Experiment

We conducted extensive experiments and in-depth analyses to evaluate the proposed entire framework, including the automatic reasoning data creation framework FMAD, the created dataset MMATH-Data, the reward model MRM and math-reasoning-oriented MMATH-LLM.

5.1 Datasets

We evaluated our approach on two widely-adopted mathematical reasoning benchmarks: GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). For GSM8K, we employed its standard test set comprising 1,319 problems across both verification and RL tasks. For MATH, following (Lightman et al., 2023), we utilized MATH-500, a curated subset of 500 representative problems from the original test set. All experiments were conducted using the LLaMA-3-8B-Instruct model.

5.2 Baselines

We compared our method against two types of models with 7B-70B parameters.

Pre-trained models. (1) LLaMA-series: We compared with LLaMA-2 (Touvron et al., 2023) and LLaMA-3 (Dubey et al., 2024). (2) Qwen2-7B-Instruct (Yang et al., 2024): an instruction-tuned model with SwiGLU activation, attention QKV bias, and group query attention. (3) Mistral-7B-Instruct-v0.3 (Jiang et al., 2023): A widely-used open-source model known for its efficient performance and instruction-following capabilities.

Fine-tuned models. (1) LLEMMA (Azerbayev et al., 2023): a specialized mathematics language model initialized with Code LLaMA weights and trained on the Proof-Pile-2 corpus for 200 billion tokens. (2) MetaMATH (Yu et al., 2023): a model built upon the LLaMA-2 architecture, employing a novel data augmentation strategy that rewrites mathematical questions from multiple perspectives without requiring external knowledge, significantly enhancing its mathematical reasoning capabilities.

Model	Verifier	GSM8K (%)	MATH (%)
LLaMA-2-7B (Touvron et al., 2023)	-	14.6	2.5
	MAD	15.1 (+0.6)	3.4 (+0.9)
	MRM (Ours)	16.4 (+1.8)	3.9 (+1.4)
LLaMA-3-8B (Dubey et al., 2024)	-	78.8	24.0
	MAD	80.0 (+1.2)	24.9 (+0.9)
	MRM (Ours)	81.2 (+2.4)	25.8 (+1.8)
Qwen2-7B (Yang et al., 2024)	-	78.4	42.9
	MAD	79.4 (+1.0)	44.9 (+1.8)
	MRM (Ours)	79.9 (+1.5)	45.5(+2.6)
Mistral-7B (Jiang et al., 2023)	-	50.6	10.2
	MAD	52.5 (+1.9)	13.0 (+2.8)
	MRM (Ours)	53.0 (+2.4)	14.6 (+4.4)

Table 2: Results comparing two verifiers MAD and our MRM across different baselines.

(3) WizardMATH (Luo et al., 2023): a reinforcement learning-based model supervised by both PRM and IRM during training. (4) BackMATH (Zhang and Xiong, 2025): a model augmented with backward reasoning data and supervised by both PRM and Backward reasoning Reward Models (BackPRM) during reinforcement learning. (5) DeepSeekMath (Shao et al., 2024): a model that is initialized from DeepSeek-Coder-v1.5-7B, and then continuously pretrained on 500 billion tokens of math-related data from Common Crawl, supplemented with natural language and code data.

5.3 Settings

We used LLaMA-3-8B-Instruct to build our MMATH-LLM and MRM. We trained MMATH-LLM for 3 epochs on the combined training sets of GSM8K and MATH, employing a learning rate of 2e-5 with cosine decay scheduling. For the MRM model, we conducted training for 2 epochs using the MMATH-Data, with a learning rate of 1e-5 and cosine decay scheduling.

5.4 Baseline Models with MRM

Table 2 present experiment results of the baselines with two different verifiers: MAD and our MRM. The results reveal consistent accuracy improvements across all evaluated models, with MRM outperforming MAD by an average of +0.8% on GSM8K and +1.1% on MATH. Notably, MRM achieves the most significant gains on Mistral-7B (+4.4% on MATH), demonstrating its effectiveness in enhancing reasoning capabilities across diverse model architectures. Compared to MAD, MRM introduces two key advancements: (1) step-level feedback granularity that improves error localization precision, and (2) debate history integration that increases correction success rates. These enhancements show varying effectiveness across models. While advanced models like Qwen2-7B achieve substantial improvements (42.9% → 45.5%), LLaMA-2-7B exhibits modest gains (2.5% → 3.9%). This performance disparity suggests that MRM's effectiveness is positively correlated with base model capacity.

Model	# Parmeters	GSM8K (%)	MATH(%)
LLaMA-2	7B	14.6	2.5
	13B	28.7	3.9
	70B	56.8	13.5
LLaMA-3	8B	78.8	24.0
Qwen2	7B	78.4	42.9
Qwen2-math	7B	-	38.0
Qwen2.5-math	7B	-	42.8
Mistral	7B	50.6	10.2
LLEMMA	7B	36.4	18.0
	34B	51.5	25.0
MetaMATH	7B	66.5	19.8
WizardMATH	7B-v1.0	54.9	10.7
	7B-v1.1	82.1	31.9
BackMATH	7B	68.1	21.9
DeepSeekMath	7B	79.9	41.0
Ours	8B	**83.4**	**45.1**

Table 3: Comparison results of our model against a wide range of baselines.

5.5 Main Results

Table 3 presents a comprehensive comparison of our proposed model against baselines on both the GSM8K and MATH datasets. Our evaluations involve various model scales, ranging from 7B to 70B parameters. Our model achieves 83.4% on GSM8K and 45.1% on MATH, outperforming all

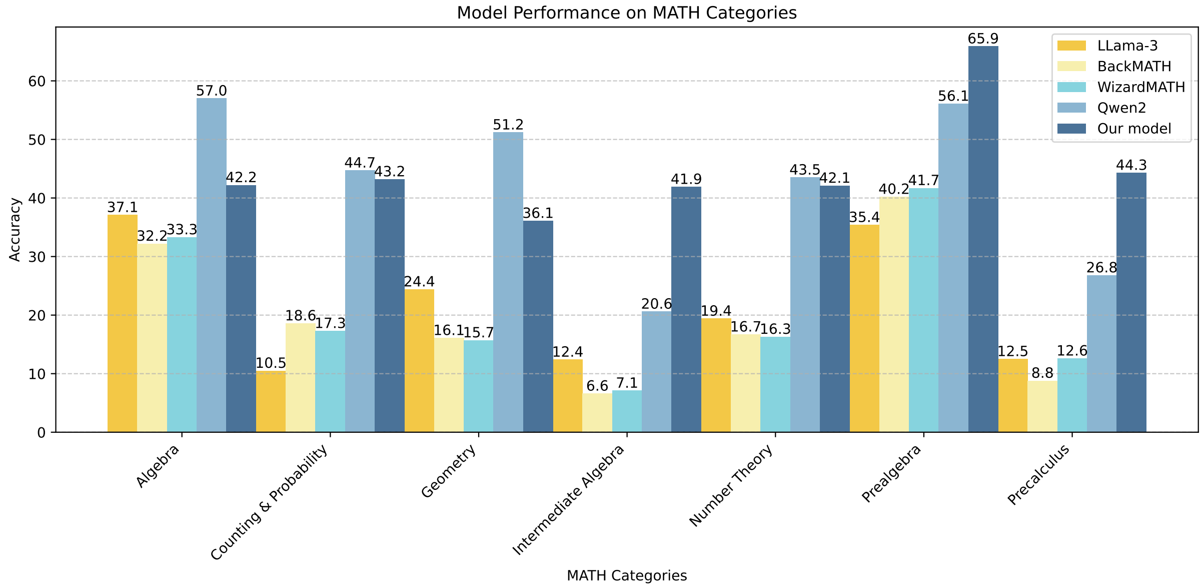


Figure 3: Experiment results on the MATH dataset.

base models and fine-tuned variants.

Comparison with Base Models: Compared to the strongest base model, Qwen2-7B (78.4% on GSM8K, 42.9% on MATH), our approach improves performance by +5.0% and +2.2%, respectively. In comparison to LLaMA-3-8B (76.9% on GSM8K, 24.0% on MATH), our model achieves +6.5% and +21.1% gains, demonstrating the effectiveness of our training methodology.

Comparison with Fine-Tuned Models: Our model surpasses WizardMath-7B-v1.1 (82.1% on GSM8K, 31.9% on MATH) by +1.3% and +13.2%, respectively, despite using a similar parameter scale. Our model also outperforms WizardMath-7B-v1.1 (82.1% → 83.4% on GSM8K, 31.9% → 45.1% on MATH) due to MRM's unified reward mechanism, which resolves the conflicting optimization objectives between PRM and IRM in WizardMath. Specifically, WizardMath's reward approach (PRM+IRM) creates inconsistent supervision signals, PRM focuses on step correctness while IRM emphasizes instruction quality, leading to suboptimal policy. Compared to DeepSeekMath-7B (79.9% on GSM8K, 41.0% on MATH), we achieve +3.5% and +4.1% improvements.

Compared to conventional fine-tuning approaches, our framework introduces two key innovations that significantly enhance mathematical reasoning capabilities: (1) FMAD, which enriches the reasoning process by annotating each step with comprehensive debate history and judge decisions, thereby improving the quality and diversity of reasoning data through multi-perspective analysis, and (2) MRM, which provides more precise and reliable supervision through fine-grained, step-level scoring, enabling better guidance of the RL process. Instead of fine-tuning the model to output LaTeX directly, we use a prompt-based approach (Appendix B) to guide the model, preserving its generalization ability while ensuring output format consistency.

5.6 Detailed Analysis on the MATH Dataset

We conducted in-depth analysis on performance of compared models for each math category on the MATH dataset. We compared four models (LLaMA-3, BackMATH, WizardMATH and our model) across several MATH categories, using accuracy as the evaluation metric. Results are shown in Figure 3. As illustrated, our model shows notable improvements in these categories, achieving 42.2% accuracy in Algebra, 36.1% in Geometry, and 65.9% in Prealgebra, demonstrating its ability to handle diverse mathematical tasks. These gains are primarily due to the MRM, which identifies and corrects computational errors through structured debate. While the improvement in Geometry is less pronounced, this is likely due to the challenges in integrating geometric intuition with algebraic reasoning. Overall, our model exhibits consistent and significant performance gains across all categories, highlighting its effectiveness in complex mathematical reasoning.

These results support our core hypothesis: the

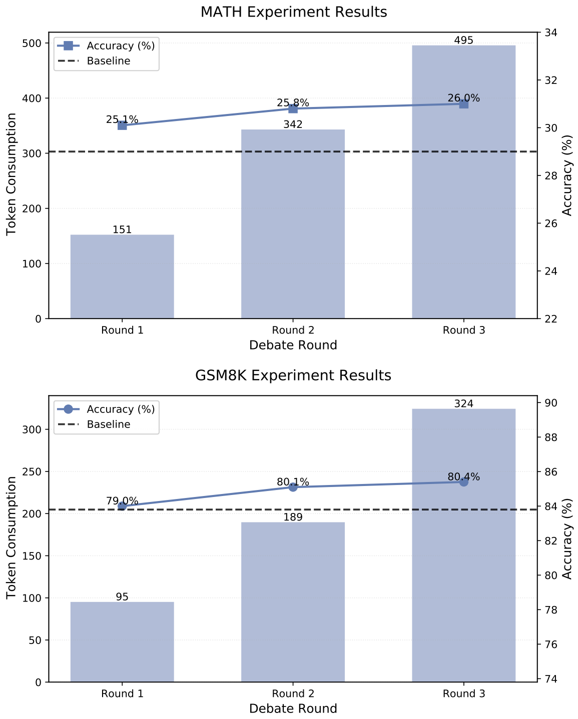


Figure 4: Token costs and accuracy of MRM on MATH and GSM8K.

debate-driven paradigm MRM enhances performance in (1) computation-intensive tasks by localizing errors, and (2) pattern-driven tasks through structured reasoning supervision. In conclusion, our model outperforms existing methods across most categories, demonstrating superior reasoning ability and making it a better choice for addressing challenging mathematical tasks.

5.7 Token Cost

Figure 4 analyzes the impact of debate rounds ($N = 1, 2, 3$) on computational cost and accuracy for mathematical reasoning. The results demonstrate a clear trade-off between performance gains and resource consumption as N increases. Specifically: A single-round debate ($N = 1$) achieves baseline improvements (MATH: 24.0% $\rightarrow$ 25.1%, +1.1; GSM8K: 78.8% $\rightarrow$ 79.0%, +0.2) with minimal token cost, but may limit debaters' the depth of reasoning due to insufficient argument exchange. A two-round debate ($N = 2$) yields substantial gains (MATH: 25.1% $\rightarrow$ 25.8%, +0.7; GSM8K: 79.0% $\rightarrow$ 80.1%, +1.1) with moderate computational overhead, striking an optimal balance between depth and efficiency. A three-round debate ($N = 3$) shows diminishing returns (MATH: 25.8% $\rightarrow$ 26.0%, +0.2; GSM8K: 80.1% $\rightarrow$ 80.4%, +0.3), despite incurring higher token costs than $N = 2$, suggesting potential information redundancy in extended debates. These results suggest that while $N = 1$ may constrain reasoning depth, additional rounds beyond $N = 2$ provide limited accuracy benefits relative to their computational expense. The marginal utility decrease beyond $N = 2$ suggests that it represents the optimal cost-performance configuration for our framework.

Model	Verifier	MATH-500(%)
LLaMA-3-8B	-	24.0
	ORM	27.4
	PRM	30.6
	Math-Shepherd	31.9
	MRM (Ours)	25.8
LLaMA-3-8B+RL	ORM	28.0
	PRM	35.4
	Math-Shepherd	42.9
	MRM (ours)	**45.1**

Table 4: Comparison of reward models with accuracy on the MATH dataset.

5.8 Comparison of Reward Models

In this section, we provide a comprehensive comparison of reward models including ORM, PRM, Math-Shepherd and our MRM. As shown in Table 4, we evaluate the models based on their performance on the MATH-500 dataset. The results indicate that, among the different reward models, the MRM consistently outperforms others, especially after reinforcement learning training. Specifically, MRM achieves the highest accuracy of 45.1%, surpassing both Math-Shepherd (42.9%) and PRM (35.4%) by a significant margin. This improvement is attributed to MRM's step-wise feedback mechanism, which effectively guides the model through complex reasoning processes. In contrast, ORM (28.0%) struggles with multi-step reasoning problems due to its inability to assess intermediate reasoning quality, while PRM (35.4%) improves over ORM but fails to resolve ambiguous steps without contextual reasoning. Math-Shepherd, which uses a Monte Carlo method for data annotation, performs well with an accuracy of 31.9%. However, the Monte Carlo method requires a high sampling density for accurate probability estimation, which can lead to less reliable annotations, potentially impacting the quality of the data used for training.

In summary, our MRM demonstrates superior performance across all categories by providing more precise and effective feedback throughout the reasoning process.

6 Conclusion

In this paper, we have presented FMAD, MRM, MMATH-Data, and MMATH-LLM. FMAD evaluates the contribution of each reasoning step in a problem using the Multi-Agent Debate framework, while MRM, trained with MMATH-Data, supervises the RL process by providing step-wise feedback. MMATH-Data, constructed by FMAD, supports MRM in guiding MMATH-LLM, a model designed specifically for mathematical reasoning, during the RL process. Through extensive experiments on the GSM8K and MATH benchmarks, we demonstrate that MRM significantly enhances the RL process. MMATH-LLM, supervised by MRM, outperforms existing methods, achieving an accuracy of 83.4% on GSM8K and 45.1% on MATH.

Limitations

First, the multi-round debate mechanism in FMAD and MRM incurs significant token consumption, especially when evaluating performance or annotating data using LLM APIs. This scalability challenge may limit practical applications in resource-constrained scenarios. Second, as shown in Figure 3, the model exhibits relatively lower performance in Geometry compared to other categories (e.g., Algebra or Number Theory). This limitation stems from its inability to integrate textual reasoning with spatial or visual information, such as interpreting geometric diagrams or coordinate systems from text descriptions. Enhancing the model's multimodal reasoning capabilities remains an important direction for our future work.

Acknowledgments

The present research was supported by the National Key Research and Development Program of China (Grant No. 2024YFE0203000). We would like to thank the anonymous reviewers for their insightful comments.

References

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2023. Llemma: An Open Language Model For Mathematics. *Preprint*, arXiv:2310.10631.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 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. *Preprint*, arXiv:2305.14325.

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*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. *arXiv preprint arXiv:2103.03874*.

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*.

Levente Kocsis and Csaba Szepesvári. 2006. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer.

Zongwei Li, Lianghao Xia, Hua Hua, Shijie Zhang, Shuangyang Wang, and Chao Huang. 2025. DiffGraph: Heterogeneous Graph Diffusion Model. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, WSDM '25, page 40–49, New York, NY, USA. Association for Computing Machinery.

Zongwei Li, Lianghao Xia, and Chao Huang. 2024. RecDiff: Diffusion Model for Social Recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, page 1346–1355, New York, NY, USA. Association for Computing Machinery.

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

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's Verify Step by Step. *arXiv preprint arXiv:2305.20050*.

Yan Liu, Renren Jin, Ling Shi, Zheng Yao, and Deyi Xiong. 2024. FineMath: A Fine-Grained Mathematical Evaluation Benchmark for Chinese Large Language Models. *Preprint*, arXiv:2403.07747.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. WizardMath: Empowering Mathematical Reasoning for Large Language Models via Reinforced Evol-Instruct. *arXiv preprint arXiv:2308.09583*.

OpenAI. 2023. GPT-4 Technical Report. *arXiv e-prints*, pages arXiv–2303.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing Mathematical Reasoning Abilities of Neural Models. *arXiv preprint arXiv:1904.01557*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models.

Tianhao Shen, Renren Jin, Yufei Huang, Chuang Liu, Weilong Dong, Zishan Guo, Xinwei Wu, Yan Liu, and Deyi Xiong. 2023. Large Language Model Alignment: A Survey. *Preprint*, arXiv:2309.15025.

Dan Shi, Tianhao Shen, Yufei Huang, Zhigen Li, Yongqi Leng, Renren Jin, Chuang Liu, Xinwei Wu, Zishan Guo, Linhao Yu, Ling Shi, Bojian Jiang, and Deyi Xiong. 2024. Large Language Model Safety: A Holistic Survey. *Preprint*, arXiv:2412.17686.

Ling Shi and Deyi Xiong. 2025. CRiskEval: A Chinese Multi-Level Risk Evaluation Benchmark Dataset for Large Language Models. In *The 63rd Annual Meeting of the Association for Computational Linguistics*.

Qwen Team. 2024. Qwen2.5: A Party of Foundation Models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 Technical Report. *Preprint*, arXiv:2407.10671.

Lei Yang, Renren Jin, Ling Shi, Jianxiang Peng, Yue Chen, and Deyi Xiong. 2025. ProBench: Benchmarking Large Language Models in Competitive Programming. *Preprint*, arXiv:2502.20868.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models. *arXiv preprint arXiv:2309.12284*.

Shaowei Zhang and Deyi Xiong. 2025. BackMATH: Towards Backward Reasoning for Solving Math Problems Step by Step. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 466–482, Abu Dhabi, UAE. Association for Computational Linguistics.

Hattie Zhou, Azade Nova, Hugo Larochelle, Aaron Courville, Behnam Neyshabur, and Hanie Sedghi. 2022. Teaching Algorithmic Reasoning via Incontext Learning. *arXiv preprint arXiv:2211.09066*.

Pengyun Zhu, Long Wen, Jinfei Liu, Feng Xue, Jian Lou, Zhibo Wang, and Kui Ren. 2023. Capp-130: A corpus of chinese application privacy policy summarization and interpretation. In *Advances in Neural Information Processing Systems*, volume 36, pages 46773–46785. Curran Associates, Inc.

Pengyun Zhu, Chaowei Zhang, Xiaofeng Li, Jifu Zhang, and Xiao Qin. 2022. A high-dimensional outlier detection approach based on local coulomb force. *IEEE Transactions on Knowledge and Data Engineering*, 35(6):5506–5520.

A Prompts of Debate

We provide detailed prompt examples for the Fine-grained Multi-Agent Debate framework:

- Appendix A.1 presents the debate rules and guidelines.
- Appendix A.2 contains the specific prompt template for Debater A.
- Appendix A.3 shows the corresponding prompt template for Debater B.
- Appendix A.4 details the prompt structure for the Judge role.

These comprehensive prompt examples demonstrate the implementation details of our debate framework.

A.1 Debate Rules

Table 5 provides debate rules in debate process, which are incorporated into prompts fed into the two debaters.

- The following are the rules for the Mathematical Debate task.

- There are three players: 2 debaters (Debater_A, Debater_B) and a Judge. Each debater is given an answer to argue.

- You will be sent a math problem and some solutions of it. Debater_A's position is to prove the given solution step is correct or beneficial for the reasoning step, while Debater_B's position is to prove the given solution step is wrong or not beneficial.

- The Debaters' goal is to win the debate by persuading the Judge that their answer is correct.

- There are a fixed number of rounds. In each round, the debaters will make arguments for why their position is correct and the other's is incorrect.

Table 5: Debate rules.

A.2 Prompt for Debater A

description

You are Debater_A, an intelligent AI with the goal of winning the debate.

Welcome to the Mathematical Reasoning Debate.

- You are concise and clearly state your arguments.
- Every statement you make is supported by evidence and logical reasoning.
- You will be sent a math problem and some solutions of it.

Your position is to prove the given solution step is correct or beneficial for the reasoning process and can reach the final answer.

- Here is the Math Problem:

——<PROBLEM>——

- Here is the Answer:

——<ANSWER>——

- Here are the Reasoning Steps:

——<RS>——

- Here is the step you should prove that it is right or has benefit for reasoning:

——<Single Step>——

- Debater_B will prove that the <Single Step> is wrong or has no benefit for reasoning.

start debate

Now it is your turn as Debater_A in ROUND <NUMBER>.

Remember to both criticize the arguments made by Debater_B and build up your own arguments.

Remember that it is your job to defend the position that <Single Step> is the correct answer and that Debater_B's position is wrong.

You should assemble the math problem as well as the context of the reasoning steps for your response, and if there are references to a particular step, include them with <quote> <\quote>.

And your response is limited to 200 tokens.

This is what your opponent, debater_B, said during their speech:<OPPONENT>

This is what you, Debater_A, said during your previous speech:<PREVIOUS_SPEECH>

Write out your debate:

Table 6: Prompts for Debater A.

A.3 Prompt for Debater B

description

You are Debater_B, an intelligent AI with the goal of winning the debate. Welcome to the Mathematical Reasoning Debate.

- You are concise and clearly state your arguments.
- Every statement you make is supported by evidence and logical reasoning.
- You will be sent a math problem and some solutions of it.

Your position is to prove the given solution step is wrong or has no benefit for the reasoning process.

- Here is the Math Problem:

—–<PROBLEM>—–

- Here are the Reasoning Steps:

—–<RS>—–

- Here is the Answer:

—–<ANSWER>—–

- Here is the step you should prove that it is wrong or has no benefit for reasoning:

—–<Single Step>—–

- Debater_A will prove that the <Single Step> is correct or has benefit for reasoning.

start_debate: Now it is your turn as Debater_B in ROUND <NUMBER>.

- Remember to both criticize the arguments made by Debater_A and build up your own arguments.

Remember that it is your job to defend the position that <Single Step> is wrong and that Debater_A's position is incorrect.

You should assemble the math problem as well as the context of the reasoning steps for your response, and if there are references to a particular step, include them with <quote> <\quote>.

And your response is limited to 200 tokens. This is what your opponent, debater_A, said during their speech:<OPPONENT>

This is what you, Debater_B, said during your previous speech:<PREVIOUS_SPEECH>

Write out your debate:

Table 7: Prompts for Debater B.

A.4 Prompt for Judge

Now it is the time for a decision. You are Judge, an intelligent AI with the goal of determining which position in the debate is correct.
- You must make a decision, as it is important for the successful completion of this mathematical reasoning task.
- Here is the Math Problem:
——<PROBLEM>——
- Here are the Reasoning Steps:
——<RS>——
- Here is the Answer:
——<ANSWER>——
- Here is the step that Debater_A and Debater_B should prove that whether it has benefit for reasoning:
——<Single Step>——
- Debater_A will prove that the <Single Step> is correct or has benefit for reasoning.
- Debater_B will prove that the <Single Step> is wrong or has no benefit for reasoning.
- Here is what Debater_A said during debate:
<Debater_A_round>
- Here is what Debater_B said during debate:
<Debater_B_round>
- Keep your answer short and concise. Your response is limited to 200 tokens.
- There is a true, correct answer.
- Consider the debaters' correctness of reasoning steps and whether the claims they made support their position.
- Do not trust any claim made by a debater unless it points out key reasoning steps.
Remember that the Debater_A's position is to prove the given solution step is correct or beneficial for the reasoning process and can reach the final answer, and Debater_B's position is to prove the given solution step is wrong or has no benefit for the reasoning process.
Please give your answer in this format: [Debater_name] [Confidence_score]
- Example 1 - Debater_X | score%
- Example 2 - Debater_X | score%
- Do not contain any other text at all.
Now give your justification:

Table 8: Prompt for Judge.

B Prompt for Solution Generation

"Below is an instruction that describes a task. "
"Write a response that appropriately completes the request."
"Write your answer with format: ANSWER:\\boxed{your answer here}"
"Instruction: instruction Response: Let's think step by step."

Table 9: Prompt for Solution Generation.