

# LLM Reasoning Engine: Specialized Training for Enhanced Mathematical Reasoning

**Shuguang Chen**

Department of Mathematics  
Purdue University  
chen4914@purdue.edu

**Guang Lin**

Department of Mathematics  
Purdue University  
guanglin@purdue.edu

## Abstract

Large Language Models (LLMs) have shown remarkable performance in various natural language processing tasks but face challenges in mathematical reasoning, where complex problem-solving requires both linguistic understanding and mathematical reasoning skills. Existing approaches to address this challenge often rely on ensemble methods and suffer from the problem of data scarcity in target domains. In this work, we present a novel method to enhance the capabilities of LLMs in mathematical reasoning tasks. Motivated by the need to bridge this gap, our approach incorporates a question paraphrase strategy, which aims to diversify the linguistic forms of mathematical questions to improve generalization. Additionally, specialized training objectives are employed to guide the model’s learning process, focusing on enhancing its understanding of mathematical concepts and reasoning processes. We conduct experiments on four datasets using different LLMs, and demonstrate the effectiveness of our approach in improving LLMs’ performance on mathematical reasoning tasks. Our findings underscore the significance of our methodology in advancing LLMs and their potential implications for real-world applications that require mathematical reasoning abilities.

## 1 Introduction

In recent years, Large Language Models (LLMs) (Touvron et al., 2023b; Team et al., 2023) have emerged as powerful tools in the field of machine learning, demonstrating remarkable performance in a wide range of downstream benchmarks. Their ability to understand and generate natural language text has revolutionized various applications, from language translation (Costa-jussà et al., 2022; Bar-rault et al., 2023) to question answering systems (Chowdhery et al., 2023; Chen et al., 2023). Central to their success is their ability to perform com-

plex reasoning, enabling them to tackle complex problems with impressive accuracy and efficiency.

However, while LLMs excel in many domains, they face challenges when it comes to solving mathematical problems. Mathematical reasoning often requires intricate logical operations and a deep understanding of mathematical concepts (Saxton et al., 2019; Lightman et al., 2023), posing significant hurdles for conventional LLM architectures. Moreover, the scarcity of data in the mathematical domain (Liu et al., 2021; Kumar et al., 2022) further compounds these challenges, limiting the performance and generalization of these LLMs.

Some existing methods (Shen et al., 2023; Jiang et al., 2024) attempt to address these challenges by employing ensemble techniques, where multiple LLMs collaboratively solve mathematical problems. However, these methods can still be further improved, particularly in enhancing the performance of individual LLMs and mitigating performance degradation over extended reasoning steps. A major challenge when applying LLMs to complex mathematical problems is error propagation, especially in tasks requiring long reasoning chains. Figure 1 illustrates this issue with an example from the GSM8K dataset (Cobbe et al., 2021), solved by the Llama2-7B model (Touvron et al., 2023b). The problem involves multiple intermediate steps to reach the correct solution. However, the model makes an error in the initial steps, which cascades through subsequent stages of reasoning. This occurs because each reasoning step depends on the accuracy of preceding steps; even a minor initial error can compound, ultimately leading to significant deviations from the correct solution. This phenomenon highlights the limitations of current LLMs in maintaining accuracy across long reasoning sequences and underscores the need for improved methodologies to mitigate error propagation and enhance performance in mathematical tasks.

In this paper, we propose novel approaches to ad-

**Question:**

Linus works for a trading company. He buys a mobile device for \$20 and sells it for twice the amount of the original price. If he bought 2 devices last Monday and 4 devices last Tuesday, how much profit was he able to earn after selling all the mobile devices he bought last Monday and Tuesday?

**Answer:**

Step 1: The total cost for the devices he bought on Monday is  $2 \times 20 = 40$   
Step 2: The total cost for the devices he bought on Tuesday is  $4 \times 20 = 80$   
Step 3: The total cost of these devices is  $80 + 40 = 120$   
Step 4: Linus sold each mobile device he bought for  $20 \times 2 = 40$   
Step 5: So, he earned  $40 \times 2 = 80$  for the two devices he bought last Monday.  
Step 6: And he earned  $40 \times 4 = 160$  for the 4 devices he bought last Tuesday.  
Step 7: The total amount of sales he made were worth  $160 + 80 = 240$   
Step 8: Linus earned  $240 - 120 = 120$   
Step 9: The answer is 120

**Model Prediction:**

Step 1: Linus was able to sell each device bought on Monday for  $20 \times 2 = \$40$  each.  
Step 2: So, his earning is  $40 - 20 = \$20$  after selling each device he bought on Monday.  
Step 3: And he was able to sell each device bought on Tuesday for  $40 \times 2 = \$80$  each.  
Step 4: So, his earning after selling each device bought on Tuesday is  $80 - 40 = \$40$ .  
Step 5: And the total earnings after selling 4 devices bought on Tuesday was  $40 \times 4 = \$160$ .  
Step 6: Hence, his total profit is  $40 + \$160 = \$200$   
Step 7: The answer is 200

Figure 1: Llama2-7B model prediction on an example from the GSM8K dataset.

dress the limitations of existing methods in solving mathematical problems with LLMs. Our contributions include the introduction of new training objectives designed to uncover underlying patterns in data, thus improving model performance. Additionally, we leverage data augmentation techniques to maximize the utility of existing datasets to fine-tune LLMs, enhancing their effectiveness in mathematical reasoning tasks. To evaluate the efficacy of our proposed methods, we conduct experiments using four open-source LLMs—Llama (Touvron et al., 2023a), Llama2 (Touvron et al., 2023b), Mistral (Jiang et al., 2023), and Mixtral (Jiang et al., 2024)—on four widely used mathematical reasoning datasets: GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), GSM8K\_Hard (Gao et al., 2022), and SVAMP (Patel et al., 2021). Our results demonstrate significant performance improvements, underscoring the effectiveness of our methods in advancing the mathematical reasoning capabilities of LLMs.

In summary, this paper presents novel contributions towards improving the performance of LLMs in solving mathematical problems, addressing the challenges posed by complex reasoning and data scarcity. Our findings have implications for advancing the field of machine learning and expanding the applicability of LLMs to various problem domains.

## 2 Background

The intersection of machine learning and mathematical reasoning has received significant attention from researchers seeking to enhance the capabilities

of LLMs in solving complex mathematical problems. In this section, we review the previous literature exploring various approaches and methodologies to address the challenges posed by mathematical reasoning tasks and highlight the advances made in this domain.

### 2.1 Recent Advances in LLMs

The evolution of LLMs represents a significant milestone in the field of natural language processing (NLP) and machine learning. Over the past decade, advances in neural network architectures, coupled with the availability of large amounts of text data, have driven the development of increasingly complex and capable LLMs. Beginning with seminal works such as Google’s BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) and OpenAI’s GPT (Generative Pre-trained Transformer) (Radford et al., 2019), researchers have made substantial progress in enhancing LLMs’ language understanding and generation capabilities (Liu et al., 2019; Raffel et al., 2020). These models leverage transformer architectures and self-attention mechanisms to capture long-range dependencies and contextual information, enabling them to generate coherent and contextually relevant text.

Further refinements, such as the introduction of GPT-3 (Brown et al., 2020) with significantly larger parameter sizes and more sophisticated training regimes, have pushed the limits of LLM performance to unprecedented levels. These advancements have paved the way for LLMs to excel in

a wide range of NLP tasks, including language translation (Costa-jussà et al., 2022; Kudugunta et al., 2024), text summarization (Lewis et al., 2019; Zhang et al., 2020), and question answering (Sanh et al., 2019; He et al., 2020).

## 2.2 LLMs for mathematics

The application of LLMs to mathematical reasoning tasks represents a recent and growing area of research within the machine learning community. Although LLMs have demonstrated remarkable proficiency in natural language understanding, their performance in mathematical reasoning tasks has historically lagged behind.

However, recent studies (Gou et al., 2023; Shao et al., 2024) have shown promising results in using LLMs to solve mathematical problems. Transfer learning techniques, particularly fine-tuning pretrained LLMs on mathematical datasets, have emerged as effective strategies for enhancing LLMs’ mathematical reasoning capabilities. By leveraging the knowledge encoded in pre-trained language models and adapting it to mathematical domains, researchers have achieved remarkable results on mathematical tasks.

Moreover, novel adaptation techniques, such as question paraphrase methods (Yu et al., 2023) and tailored training objectives (Liu et al., 2023) aimed at enhancing LLMs’ understanding of mathematical reasoning processes, have further advanced LLMs in mathematical applications. These techniques enable LLMs to effectively leverage existing data and develop robust reasoning abilities, thus expanding their utility in mathematical problem solving scenarios. However, these existing methods suffer from the problem of error propagation over long reasoning paths.

Overall, the application of LLMs in mathematics has immense potential to revolutionize the way mathematical problems are approached and solved. As research in this area continues to evolve, we can expect further advancements in LLMs’ mathematical reasoning capabilities and their integration into diverse mathematical domains.

## 3 Methodology

The motivation for our methodology arises from the observation that while LLMs excel in various natural language processing tasks, their performance in mathematical reasoning remains suboptimal. This gap is primarily due to the inherent complexity of

mathematical problems, which often requires intricate reasoning and logical deduction (Saxton et al., 2019; Lightman et al., 2023). Furthermore, the limited availability of annotated data in the mathematical domain (Liu et al., 2021; Kumar et al., 2022) presents a significant obstacle to effectively training LLMs for such tasks.

To overcome these challenges, we propose a novel approach that combines question paraphrasing techniques with tailored training objectives to strengthen the mathematical reasoning capabilities of LLMs. An overview of our proposed pipeline is illustrated in Figure 2.

### 3.1 Question Paraphrase

Question paraphrasing is a crucial technique employed to augment existing data and enhance the model’s ability to generalize across different linguistic forms of mathematical problems. This process involves leveraging the powerful generative capabilities of the GPT-4 model (Brown et al., 2020) to produce diverse paraphrases for each question in the dataset. By generating multiple variations of the same question while preserving its semantic meaning, we aim to enrich the training data and expose the model to a wider range of linguistic structures and expressions commonly encountered in mathematical problem-solving scenarios.

**Paraphrasing Questions** The GPT-4 model is utilized to generate the paraphrases for each question in the data set. Given a mathematical question, the model generates alternative phrasings that convey the same underlying mathematical concept. This step significantly enhances the diversity of the training data by presenting questions in various linguistic forms, such as synonyms, paraphrases, and syntactic variations.

To ensure the quality and semantic coherence of the paraphrased questions, each paraphrase, along with its corresponding answer, is inputted into the GPT-4 model. The model is then tasked with determining whether the paraphrased question-answer pairs match or not. This iterative validation process helps filter out the inconsistencies or semantic distortions, ensuring that only high-quality paraphrases are retained for training.

### 3.2 Special Training Objectives

Effective training objectives are critical for guiding a model’s learning process and fostering a deeper understanding of mathematical concepts and rea-

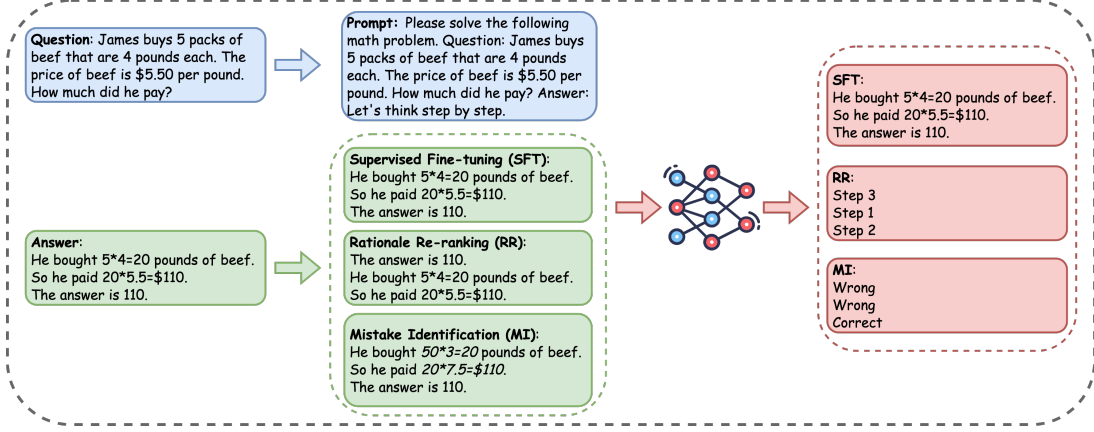


Figure 2: The overview of our proposed pipeline.

soning. Our approach incorporates specialized training objectives designed to address the unique challenges of mathematical reasoning tasks.

During training, we start with a pretrained LLM and apply a supervised fine-tuning (SFT) objective on a carefully curated dataset of mathematical problems. This data set includes a diverse range of questions and solutions to ensure a comprehensive coverage of mathematical concepts and problem types. In the SFT phase, the model is trained to minimize the loss between its predicted answers and the correct solutions provided in the dataset, effectively aligning its outputs with the desired responses.

**Rationale Re-Ranking (RR)** The Rationale Re-Ranking (RR) objective aims to improve the model’s ability to identify and reconstruct the correct reasoning path for solving mathematical problems. This objective involves shuffling the reasoning steps associated with a given problem and reordering the models to reconstruct the proper solution sequence. By training the model to recognize and organize the logical progression of mathematical solutions, the RR objective fosters a more structured, coherent, and accurate reasoning process.

**Mistake Identification (MI)** The MI objective focuses on improving the robustness and error tolerance of the model by training it to differentiate between correct and erroneous reasoning steps. During training, random modifications are introduced to induce errors in the reasoning process, such as changing numerical values or altering logical operators. The model is then trained to distinguish between correct and erroneous reasoning steps, thus learning to identify and rectify potential mistakes. This objective helps mitigate the risk of erroneous

predictions and enhances the model’s overall performance on mathematical reasoning tasks:

Integrating these training objectives into the fine-tuning process equips LLMs with enhanced capabilities for mathematical reasoning, thereby addressing the challenges posed by complex problem-solving scenarios.

### 3.3 Training Process

The training process of our proposed method consists of several stages designed to enhance the mathematical reasoning capabilities of LLMs. The process begins with data augmentation through question paraphrasing, where GPT-4 generates various paraphrases for each mathematical question. To ensure data set quality and integrity, we verify the consistency of the paraphrased questions and their corresponding answers, iterating this process to achieve high accuracy.

Next, we introduce specialized training objectives to improve the model’s reasoning abilities. One such objective is Rationale Re-Ranking (RR), where reasoning steps are shuffled, and the model is tasked with predicting the correct sequence. This trains the model to understand and reconstruct logical progressions. Another objective is Mistake Identification (MI), where the model learns to distinguish between correct and intentionally altered reasoning steps, enhancing its ability to detect and correct errors in complex problem-solving scenarios. These objectives are incorporated within a multitask learning framework, with weighted losses assigned to each task to ensure balanced and effective training. This approach enables the model to optimize simultaneously for various aspects of mathematical reasoning, resulting in a more robust and versatile skill set.



By combining data augmentation, targeted training objectives, and a multitask learning framework, our method equips LLMs with strong mathematical reasoning skills, significantly improving their performance on challenging mathematical tasks.

**Final training objective** The final training objective is formulated as the weighted sum of  $\mathcal{L}_{SFT}$ ,  $\mathcal{L}_{RR}$ , and  $\mathcal{L}_{MI}$ :

$$\mathcal{L}_{final}(\theta) = \lambda_1 \mathcal{L}_{SFT} + \lambda_2 \mathcal{L}_{RR} + \lambda_3 \mathcal{L}_{MI}$$

where  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  are parameters that weight the importance of each loss, and  $\mathcal{L}_{SFT}$ ,  $\mathcal{L}_{RR}$ , and  $\mathcal{L}_{MI}$  are cross entropy losses.

## 4 Experiments

Mathematical reasoning challenges LLMs, requiring both natural language understanding and mathematical problem-solving skills. Evaluating the effectiveness of LLMs in mathematical reasoning is crucial for advancing state-of-the-art NLP techniques and expanding their applicability across diverse problem domains. This study seeks to assess the performance of various LLMs on mathematical reasoning tasks through targeted experiments.

### 4.1 Datasets

We conducted experiments on four datasets specifically designed to assess LLMs’ performance in mathematical reasoning.

- **GSM8K** (Cobbe et al., 2021): A comprehensive dataset comprising mathematical problems covering a wide range of topics and difficulty levels.
- **MATH** (Hendrycks et al., 2021): A curated collection of mathematical questions and solutions, designed to assess LLMs’ ability to solve mathematical problems in various domains.
- **GSM8K-Hard** (Gao et al., 2022): A subset of the GSM8K dataset containing challenging mathematical problems aimed at evaluating the robustness of LLMs under difficult scenarios.
- **SVAMP** (Patel et al., 2021): A specialized dataset focusing on mathematical reasoning in the context of symbolic mathematics, presenting unique challenges for LLMs due to its symbolic nature.

### 4.2 Base Models

We employed four base models for our experiments, each representing a distinct architecture or variant of LLMs:

- **LLama** (Touvron et al., 2023a): A baseline LLM model known for its strong performance in natural language understanding tasks.
- **LLama2** (Touvron et al., 2023b): An enhanced version of LLama, incorporating improvements in model architecture and training methodology.
- **Mistral** (Jiang et al., 2023): A state-of-the-art LLM model specifically designed for mathematical reasoning tasks, leveraging advanced adaptation techniques.
- **Mixtral** (Jiang et al., 2024): A high-capacity LLM model based on the Mixtral architecture, featuring eight times the parameter size of LLama for enhanced performance.

**Training Details** We fine-tuned these base models on the aforementioned datasets with hyperparameters tailored to each model and dataset. The parameters are set as default to the ones used during pre-training. Note that we only experiment with the version of 7B parameters for each model. We adapt LoRA (Hu et al., 2022) to make model fine-tuning more efficient. The rank and alpha are both set as 64. Additionally, we utilized specialized training objectives, such as Rationale Re-ranking (RR) and Mistake Identification (MI), to enhance the models’ understanding of mathematical reasoning. We fine-tune each model with a single Nvidia A100 GPU and the runtime of each experiment is between 2 to 6 hours.

### 4.3 Main Results

Our experiments resulted in notable performance improvements across all four base models. Table 1 presents a summary of the main results. Our proposed special training objectives yielded an average performance boost of 4.25% on GSM8K, 2.32% on MATH, 6.21% on GSM\_HARD, and 5.15% on SVAMP datasets. Moreover, combining question paraphrase with these objectives further enhances the improvement to 7.32% on GSM8K, 3.63% on MATH, 7.72% on GSM\_HARD, and 6.78% on SVAMP. Notably, our methods have a more significant impact on relatively weaker models, likely

Method	Dataset			
	GSM8K	MATH	GSM_HARD	SVAMP
<i>Baselines (SFT Only)</i>				
Llama-7B (Touvron et al., 2023a)	32.07	5.60	23.43	38.24
Llama2-7B (Touvron et al., 2023b)	36.92	5.68	26.72	41.07
Mistral-7B (Jiang et al., 2023)	58.68	14.08	55.42	50.25
Mixtral-8x7B (Jiang et al., 2024)	65.44	30.13	64.93	68.73
<i>Proposed Method (SFT + MI + RR)</i>				
Llama-7B (Touvron et al., 2023a)	37.15	6.91	29.71	44.22
Llama2-7B (Touvron et al., 2023b)	43.04	7.65	38.74	49.03
Mistral-7B (Jiang et al., 2023)	62.87	15.98	60.02	56.87
Mixtral-8x7B (Jiang et al., 2024)	67.04	34.22	66.86	68.76
<i>Proposed Method + Question Paraphrase</i>				
Llama-7B (Touvron et al., 2023a)	41.74	7.64	37.87	46.14
Llama2-7B (Touvron et al., 2023b)	46.86	9.05	32.96	53.04
Mistral-7B (Jiang et al., 2023)	65.82	17.27	63.25	58.03
Mixtral-8x7B (Jiang et al., 2024)	67.96	36.07	67.28	68.19

Table 1: Experimental results of base models on different datasets, respectively. Scores are calculated with the accuracy metric.

because these models benefit more from structured reasoning guidance. This aligns with findings in previous LLM fine-tuning studies, where weaker models exhibit larger relative improvements when exposed to specialized training objectives.

These findings underscore the empirical effectiveness of our methodology in improving the reasoning efficiency and accuracy of LLMs. By assessing LLMs’ performance in mathematical reasoning tasks, we contribute to the ongoing efforts to advance the state-of-the-art in natural language processing and pave the way for their application in diverse problem domains requiring mathematical reasoning abilities.

## 5 Analysis and Discussion

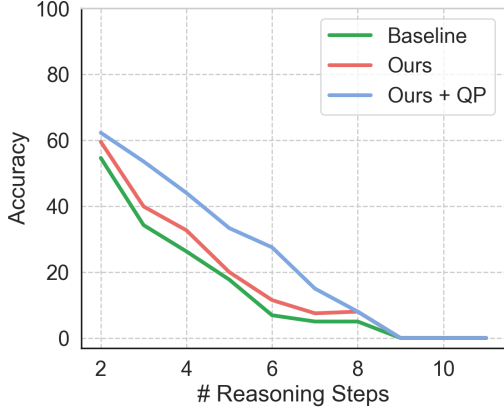
In this section, we delve into a comprehensive analysis and discussion of the experimental results, focusing on the effectiveness of our proposed method in enhancing LLMs’ capabilities in mathematical reasoning tasks. We begin by dissecting the model’s performance over varying numbers of reasoning steps, shedding light on the impact of our approach on problem-solving efficiency. Subsequently, we present findings from an ablation study aimed at elucidating the importance of individual components within our proposed methods. Finally, through a series of case studies, we illustrate both the successes and limitations of our approach, pro-

viding valuable insights for future research directions.

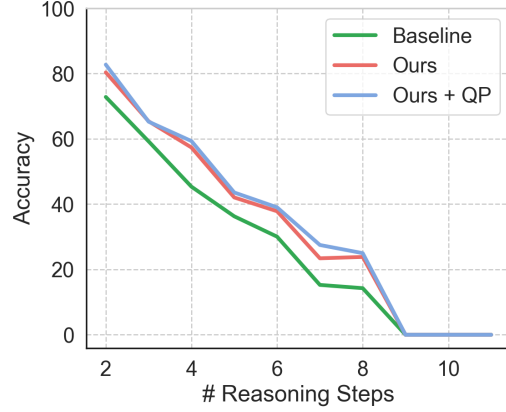
### 5.1 Analysis on Reasoning Steps

We performed an in-depth analysis of model performance across varying reasoning steps to evaluate the effectiveness of our proposed method in solving complex mathematical problems. By examining performance at different depths of reasoning, we sought to highlight the impact of our approach on problem-solving efficiency. Figure 3 summarizes the model’s performance on math questions requiring different reasoning steps. Overall, the results indicate that model performance declines as the number of reasoning steps needed to solve the problems increases. Although the improvement is minimal for questions requiring only a few reasoning steps (fewer than 4 steps), it becomes substantial for questions requiring longer reasoning chains (4–7 steps). Additionally, we note that the models struggle with questions that demand extremely long reasoning paths (more than 8 steps).

Our analysis revealed a clear trend of improved model performance as the number of reasoning steps increased. This trend indicates that our proposed method effectively enhances the model’s ability to solve complex mathematical problems, leading to more accurate solutions. Specifically, we observed substantial performance improvements



(a) Llama2-7B on different reasoning steps



(b) Mistral-7B on different reasoning steps

Figure 3: Analysis of Llama2 and Mistral on different reasoning steps, respectively. X-axis is the number of reasoning steps required to solve a math question and Y-axis the accuracy score. We use supervised fine-tuning as the baseline, and compare it with our proposed method using special training objective. QP stands for question paraphrase.

Method	GSM8K		MATH	
	Llama2-7B	Mistral-7B	Llama2-7B	Mistral-7B
Baseline (SFT)	36.92	58.68	5.68	14.08
+ RR	38.94	59.06	5.83	13.07
+ MI	40.01	61.57	7.83	15.24
+ MI + RR	43.04	62.87	7.65	15.98
+ MI + RR + QP	46.86	65.82	9.05	17.27

Table 2: Ablation Study of our proposed method on GSM8K and MATH using Llama2-7B and Mistral-7B as base models, respectively. SFT, RR, MI, and QP stand for supervised fine-tuning, rationale re-ranking, mistake identification, and question paraphrase, respectively. Scores are calculated with the accuracy metric.

on problems that involve multiple reasoning steps, which underscores the effectiveness of our approach in addressing complex problem-solving scenarios.

## 5.2 Ablation Study

To further evaluate the importance of each component in our proposed methods, we performed an ablation study in which we systematically removed individual components and evaluated the model performance. Table 2 presents an ablation study of our proposed method on GSM8K and MATH datasets using Llama2 and Mistral as base models, respectively. Our results demonstrated that all components of our proposed methods are integral to improve model performance in mathematical reasoning tasks.

Specifically, when components such as question

paraphrase techniques and specialized training objectives were integrated, we observed a marked increase in model performance, underscoring the importance of these components in facilitating effective mathematical reasoning. These findings emphasize the holistic nature of our proposed approach, wherein each component synergistically contributes to overall model performance.

## 5.3 Case Study

We present a case study in Figure 4 to demonstrate the effectiveness of our method in solving complex mathematical problems. The case study includes a positive example showcasing successful problem-solving outcomes and a negative example highlighting challenges and limitations.

The positive example illustrates how our method empowers the model to navigate intricate mathematical problems and arrive at the accurate solution, even if it does not follow the exact reasoning path provided in the answer. This example validates the effectiveness of our approach in addressing real-world mathematical challenges.

In contrast, the negative example in Figure 4 reveals scenarios where our proposed method encounters limitations or fails to produce satisfactory results. Although the model follows the correct reasoning path, a common failure mode observed was arithmetic miscalculations despite correct reasoning paths. This suggests that while LLMs grasp mathematical structure, they struggle with

<p><b>Question:</b> Micheal loves riding a bike. He rode it at least 5 times a week and makes 25 kilometers each time. He did that for four weeks, and then he decided, to ride the bike only 2 times a week, but for 60 kilometers each time, and he did that for 3 weeks. How many kilometers did Micheal do in total?</p>
<p><b>Answer:</b> Step 1: In the first part, Micheal did <math>5 * 25 = 125</math> kilometers every week. Step 2: He did that for four weeks, which means he made during that time <math>125 * 4 = 500</math> kilometers. Step 3: After that, he made <math>2 * 60 = 120</math> kilometers a week. Step 4: And Micheal did that for three weeks, so he rode <math>120 * 3 = 360</math> kilometers during that time. Step 5: In total Micheal did <math>500 + 360 = 860</math> kilometers. Step 6: The answer is 860.</p>
<p><b>Model Prediction:</b> Step 1: In four weeks, he made 5 rides per week, and he did that for 4 weeks, so he made <math>5 * 4 = 20</math> rides. Step 2: Each ride was 25 kilometers, so he made <math>25 * 20 = 500</math> kilometers in four weeks. Step 3: The next week he started making 2 rides per week, and he did that for 3 weeks, so he made <math>2 * 3 = 6</math> rides. Step 4: Each ride was 60 kilometers, so he made <math>60 * 6 = 360</math> kilometers. Step 5: In total, he made <math>500 + 360 = 860</math> kilometers. Step 6: The answer is 860.</p>

<p><b>Question:</b> Sean is practicing for his role in a theater production. He has to memorize his lines for two scenes and the lyrics to one solo song. His solo song has 54 lines in the lyrics. The first scene has twice the number of lines, but only a third of them are his lines. The second scene has six more lines than the song, and four-fifths of them are his. How many lines does Sean have to memorize?</p>
<p><b>Answer:</b> Step 1: The first scene has twice the number of lines as his solo song, so it has <math>2 * 54 = 108</math> lines. Step 2: A third of them are Sean's lines, so he has to memorize <math>108 / 3 = 36</math> lines. Step 3: The second scene has <math>54 + 6 = 60</math> lines. Step 4: Four fifths of the second scene's lines are Sean's, so he has to memorize <math>4 * 60 / 5 = 48</math> lines. Step 5: Sean has to memorize his entire solo song, so he has to memorize <math>54 + 36 + 48 = 138</math> lines in all. Step 6: The answer is 138.</p>
<p><b>Model Prediction:</b> Step 1: First find the number of lines in the first scene: <math>54 \text{ lines} * 2 = 108 \text{ lines}</math> Step 2: Then find the number of Sean's lines in the first scene: <math>108 \text{ lines} / 3 = 36 \text{ lines}</math> Step 3: Then find the number of lines in the second scene: <math>54 \text{ lines} + 6 \text{ lines} = 60 \text{ lines}</math> Step 4: Then find the number of Sean's lines in the second scene: <math>60 \text{ lines} * 4/5 = 48 \text{ lines}</math> Step 5: Then add the number of lines in each scene and the song to find the total number of lines Sean has to memorize: <math>36 \text{ lines} + 48 \text{ lines} + 54 \text{ lines} = 148 \text{ lines}</math> Step 6: The answer is 132.</p>

Figure 4: Case study on the GSM8K dataset using the Mistral-7B Model. The example on the left is positive while the example on the right is negative.

precise computation—an issue that could be mitigated by integrating external calculation modules. This example identifies areas for improvement and prompts discussions on potential future research directions, including refining methodologies, leveraging external calculation tools, or exploring math verification approaches to enhance model performance in challenging scenarios.

## 5.4 Discussion

**Data Efficiency and Generalization** Our question paraphrase strategy addresses a fundamental challenge in mathematical reasoning: the scarcity of diverse high-quality training data. By systematically transforming existing questions into linguistically varied forms while preserving their mathematical essence, we achieve improved generalization without requiring additional annotated examples. This approach is particularly valuable for specialized mathematical domains where expert annotation is costly and time-consuming. Our experimental results demonstrate that models trained with paraphrased questions exhibit enhanced robustness to linguistic variations, more closely mirroring the diverse ways in which mathematical problems may be encountered in real-world applications. This linguistic flexibility, combined with our specialized training objectives, enables models to focus on the underlying mathematical structures rather than become overly sensitive to specific phrasings. The data efficiency of our approach makes it particularly applicable to resource-constrained settings and suggests promising directions for self-supervised learning techniques that leverage mathematical invariance across different problem formulations.

**Training-focused V.S. Test-time computational methods** In recent years, the Test-Time Computation (TTC) paradigm has gained significant traction as a means to improve the reasoning skills of LLMs, particularly for complex tasks. Although training-focused and TTC methods represent different strategies, they are not mutually exclusive. In fact, their synergistic integration holds significant promise for future advancements in mathematical reasoning for LLMs. The reasoning abilities that TTC techniques leverage are fundamentally learned during the training phase, encompassing both pre-training and fine-tuning. Training on high-quality mathematical data, including examples of step-by-step reasoning, provides the essential knowledge base that enables TTC methods (e.g., CoT (Wei et al., 2022) and Self-Refine (Madaan et al., 2023)) to be effective. Furthermore, training processes can be optimized to produce models that are particularly adept at generating effective reasoning traces, which can then be further explored or verified during test time using TTC techniques. The quality of the underlying model, shaped by training, significantly influences how effectively it can utilize TTC methods.

## 6 Conclusion

In this study, we addressed the challenge of enhancing LLMs' capabilities in mathematical reasoning tasks. Our proposed method leverages innovative techniques to improve LLMs' understanding of mathematical concepts and reasoning processes. Through rigorous experimentation, we demonstrated the effectiveness of our approach in improving LLMs' performance on various mathematical problems.



Our research advances the field of natural language processing by providing a comprehensive approach to enhancing LLMs’ capabilities in mathematical reasoning tasks. By identifying key factors influencing model performance and proposing effective solutions, we bridge the gap between natural language understanding and mathematical reasoning. Our study advances LLMs’ reasoning capabilities, paving the way for broader applications in automated theorem proving, mathematical education, and AI-assisted research in scientific domains. Future work could explore integrating symbolic computation with LLMs to further enhance mathematical reliability.

## Limitations

While our study demonstrates promising results in improving the reasoning efficiency and accuracy of LLMs through data enhancement and fine-tuning, several limitations should be acknowledged. First, despite efforts to create a diverse data set through enhancement, inherent biases may persist in the training data. These biases could potentially skew the model’s performance towards certain types of mathematical problems or reasoning patterns. Second, although our fine-tuned model improves test performance, its generalization to out-of-distribution mathematical problems remains uncertain. Future work could explore curriculum learning strategies or hybrid neural-symbolic approaches to mitigate this limitation. Further investigation is needed to assess the model’s capabilities in tackling advanced mathematical concepts or interdisciplinary problems that deviate significantly from the training examples.

## References

- Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan, John Hoffman, et al. 2023. Seamlessm4t-massively multilingual & multimodal machine translation. *arXiv preprint arXiv:2308.11596*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, et al. 2023. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2022. Pal: Program-aided language models. *arXiv preprint arXiv:2211.10435*.
- Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Minlie Huang, Nan Duan, Weizhu Chen, et al. 2023. Tora: A tool-integrated reasoning agent for mathematical problem solving. *arXiv preprint arXiv:2309.17452*.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.
- 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. *NeurIPS*.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [Lora: Low-rank adaptation of large language models](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- 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*.

- Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.
- Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2024. Madlad-400: A multilingual and document-level large audited dataset. *Advances in Neural Information Processing Systems*, 36.
- Vivek Kumar, Rishabh Maheshwary, and Vikram Pudi. 2022. Practice makes a solver perfect: Data augmentation for math word problem solvers. *arXiv preprint arXiv:2205.00177*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- 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*.
- Qianying Liu, Wenyu Guan, Sujian Li, Fei Cheng, Daisuke Kawahara, and Sadao Kurohashi. 2021. Roda: Reverse operation based data augmentation for solving math word problems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:1–11.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Yixin Liu, Avi Singh, C Daniel Freeman, John D Co-Reyes, and Peter J Liu. 2023. Improving large language model fine-tuning for solving math problems. *arXiv preprint arXiv:2310.10047*.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.
- Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. [Are NLP models really able to solve simple math word problems?](#) In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. *arXiv preprint arXiv:1904.01557*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, YK Li, Y Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Sheng Shen, Le Hou, Yanqi Zhou, Nan Du, Shayne Longpre, Jason Wei, Hyung Won Chung, Barret Zoph, William Fedus, Xinyun Chen, et al. 2023. Mixture-of-experts meets instruction tuning: A winning combination for large language models. *arXiv preprint arXiv:2305.14705*.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- 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.
- Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhengguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International conference on machine learning*, pages 11328–11339. PMLR.