BackMATH: Towards Backward Reasoning for Solving Math Problems Step by Step

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 achieved impressive results in reasoning, particularly in multi-step reasoning tasks. However, when faced with more complex mathematical problems, the performance of LLMs drops significantly. To address this issue, in this paper, we propose a backward reasoning dataset, BackMATH-Data. The dataset comprises approximately 14K backward reasoning problems and 100K reasoning steps. It follows a result-oriented approach, to construct backward reasoning problems by swapping the reasoning results with specific solving conditions in the original problems. Additionally, we introduce Backward-reasoning Process-supervision Reward Model (BackPRM) and BackMATH-LLM. BackPRM supervises the quality of the generated backward reasoning problems, while BackMATH-LLM is designed for mathematical reasoning. BackMATH-LLM is fine-tuned and enhanced through reinforcement learning by supervising the quality of backward reasoning problems and by providing feedback on reasoning steps, thereby improving the mathematical reasoning capabilities of LLMs. Extensive experiments demonstrate that our model achieves an accuracy of 68.1% on the GSM8K dataset and 21.9% on the MATH dataset, exceeding the SOTA by 1.6% and 2.1% respectively.

Original problem and output of an example from GSM8K:

Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?

Answer: Natalia sold 48/2 = 24 clips in May. Natalia sold 48+24 = 72 clips altogether in April and May. 72

Backward problem and output of the example from GSM8K:

Natalia sold clips to x of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May? If we know the answer is 72, what is the value of x?

Answer: 48

(a) Backward reasoning on GSM8K

Original problem and output of an example from MATH:

Find the matrix M such that

$$M\begin{pmatrix}1 & -2\\ 1 & 4\end{pmatrix}=\begin{pmatrix}6 & 0\\ 0 & 6\end{pmatrix}$$

Output: The inverse of $\begin{pmatrix}1 & -2\\ 1 & 4\end{pmatrix}$ is

$$\frac{1}{(1)(4)-(-2)(1)}\begin{pmatrix}4 & 2\\ -1 & 1\end{pmatrix}=\frac{1}{6}\begin{pmatrix}4 & 2\\ -1 & 1\end{pmatrix}.$$

So, multiplying by this inverse on the right, we get

$$M=\begin{pmatrix}6 & 0\\ 0 & 6\end{pmatrix}\cdot\frac{1}{6}\begin{pmatrix}4 & 2\\ -1 & 1\end{pmatrix}=\boxed{\begin{pmatrix}4 & 2\\ -1 & 1\end{pmatrix}}$$

Backward problem and output of the example from MATH:

Find the matrix A such that

$$\begin{pmatrix}4 & 2\\ -1 & 1\end{pmatrix}A=\begin{pmatrix}6 & 0\\ 0 & 6\end{pmatrix}$$

Output: The inverse of $\begin{pmatrix}4 & 2\\ -1 & 1\end{pmatrix}$ is

$$\frac{1}{(1)(4)-(2)(-1)}\begin{pmatrix}1 & -2\\ 1 & 4\end{pmatrix}=\frac{1}{6}\begin{pmatrix}1 & -2\\ 1 & 4\end{pmatrix}.$$

So, multiplying by this inverse on the right, we get

$$A=\frac{1}{6}\begin{pmatrix}4 & 2\\ -1 & 1\end{pmatrix}\cdot\begin{pmatrix}6 & 0\\ 0 & 6\end{pmatrix}=\boxed{\begin{pmatrix}1 & -2\\ 1 & 4\end{pmatrix}}$$

(b) Backward reasoning on MATH

Figure 1: Examples of backward reasoning on both GSM8K and MATH.

1 Introduction

Large language models exemplified by ChatGPT and GPT-4 (OpenAI, 2022, 2023), are capable of solving tasks that require complex reasoning. Despite LLMs' outstanding performance in various domains, these models face significant challenges when solving complex mathematical problems (Saxton et al., 2019; Zhou et al., 2022). Even the most advanced models show clear deficiencies when tackling mathematical problems that require complex understanding and reasoning, often producing hallucination (Maynez et al., 2020) or exhibiting a tendency to invent facts when they are uncertain about the math problems (Bubeck et al., 2023). This limitation not only restricts the reasoning abilities of LLMs on complex mathematical problems but also highlights the urgent need for more effective strategies (Shen et al., 2023) and data augmentation techniques (Zha et al., 2023) to enhance problem-solving capabilities of LLMs.

High-quality data is instrumental in enhancing model performance (Lee et al., 2023; Shi et al., 2024; Guo et al., 2023; Huang and Xiong, 2023; Liu et al., 2024). Backward reasoning (Jiang et al., 2023), as a data augmentation technique, traces candidate answers back to the original problem to verify the presence of supporting data, thereby determining whether the model has produced hal-

*Corresponding author.

Proceedings of the 31st International Conference on Computational Linguistics: Industry Track, pages 466–482
January 19–24, 2025. ©2025 Association for Computational Linguistics

lucinations during the reasoning process. Figure 1 shows two examples of backward reasoning. Unfortunately, LLMs exhibit significant deficiencies in backward reasoning. Even provided with fullfiled prompts and demonstrations, LLMs often fail to accurately determine the backward reasoning direction when faced with complex mathematical problems. Thus, enhancing backward reasoning in LLMs is crucial for improving their ability to tackle complex tasks.

Chain-of-Thought (CoT) (Nye et al., 2021; Wei et al., 2022; Kojima et al., 2022) has been widely used to solve problems step by step. In complex reasoning tasks, CoT significantly enhances the reasoning capabilities of LLMs. In solving complex mathematical problems, compared to the Outcome Reward Model (ORM) (Christiano et al., 2017), Process-supervision Reward Model (PRM) (Uesato et al., 2022; Ziegler et al., 2019), providing feedback on reasoning steps, achieves greater accuracy and reliability on reasoning.

Inspired by *backward reasoning* and *process supervision*, in this paper, we propose BackMATH-Data, a backward reasoning dataset. This dataset is derived from mathematical problems in the training datasets of GSM8K and MATH, collected and filtered manually. ChatGPT is used to automatically generate the data instances, which are then reviewed and proofread by humans. After further reviewing and proofreading, we obtain a total of 14K backward reasoning problems with 100K reasoning steps.

Additionally, we introduce Backward-reasoning Process-supervision Reward Model (BackPRM) and BackMATH-LLM. BackPRM scores the backward reasoning steps to assess the quality of the reformulated backward reasoning problems. For BackMATH-LLM, we first perform Supervised Fine-Tuning (SFT) on the model using pairs of original and backward reasoning problems, enabling the model to construct backward reasoning problems. Subsequently, we use BackPRM and PRM to provide feedback during the reinforcement learning, where the former evaluates the quality of the backward reasoning problems while the latter provides feedback scores for each reasoning step in the solution.

In a nutshell, our contributions are listed as follows:

- We release a backward reasoning dataset that enhances model performance on complex mathematical problems. The dataset contains 14K problems and 100K reasoning steps.
- We introduce BackMATH-LLM, which effectively enhances the mathematical reasoning capabilities of LLMs and BackPRM, which provides feedback from backward reasoning on reinforcement learning to efficiently train BackMATH-LLM.
- Experiments on the GSM8k and MATH benchmarks demonstrate that our approach outperforms existing methods.

2 Related Work

2.1 Process Supervision Data

In training LLMs, high-quality data greatly optimizes the process, whereas merely expanding model size is insufficient to achieve high performance on challenging tasks like arithmetic and symbolic reasoning (Rae et al., 2021). Several studies have explored data related to process supervision. OpenAI releases the first process supervision dataset PRM800k (Lightman et al., 2023). FELM (chen et al., 2024) conducts a factual evaluation on text generated by LLMs using a custom dataset comprising 847 questions across five domains. This dataset, generated by ChatGPT, is split into individual sentences, and each reasoning step is annotated as true or false. Li et al. (2024) primarily focus on identifying erroneous steps in the reasoning process. To evaluate the honesty of LLMs, Yang et al. (2023b) annotate each reasoning step as either known or unknown. Yu et al. (2023) construct MetaMathQA, a dataset including content from the GSM8K dataset that has been rewritten using backward reasoning.

In this study, we curate the BackMATH-Data, which focuses on data in mathematics. It applies backward reasoning rules to reconstructing problems from existing datasets, particularly the MATH dataset, and generating new problems for data augmentation. Additionally, the reasoning processes of the new dataset are scored in detail.

2.2 Process Supervision

In Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017), most studies use ORM to supervise training process (Ouyang et al., 2022). However, ORM focuses solely on final results, leading to sparse rewards in end-to-end learning, which hinders reasoning supervision for complex tasks. OpenAI studies PRM and demonstrates that PRM yields better results than ORM. Luo et al. (2023) use both PRM and Instruction Reward Model (IRM) to supervise the training process.

Since there has been no PRM specifically designed for backward reasoning, our BackPRM is the first attempt in building a reward model aimed at supervising the backward reasoning process.

2.3 Fine-Tuning for Math Problem Reasoning

Fine-tuning has proven effective in enhancing LLMs' reasoning capabilities (Uesato et al., 2022; Lightman et al., 2023; Tian et al., 2023; Wu et al., 2024), particularly when it is equipped with data augmentation methods such as evol-instruct (Luo et al., 2023) and problem bootstraping (Yu et al., 2023). Among various fine-tuning approaches, current research indicates that process supervision has an advantage over outcome supervision (Lightman et al., 2023).

Inspired by process supervision and fine-tuning methods, we propose BackMATH-LLM in this paper. This model enhances the mathematical reasoning capabilities of LLMs through reinforcement learning based on feedback from backward reasoning and supervision of the reasoning steps. Our proposed model achieves higher accuracy compared to SOTA models.

3 Dataset Creation

Our key interest is to create high-quality backward reasoning problems and reasoning steps. We detail the data collection process, with a focus on the creation of data from the MATH dataset. Unlike the well-structured GSM8K dataset, which allows LLMs to directly generate backward reasoning problems based on predefined rules, the MATH dataset encompasses seven categories within mathematics (*e.g.*, algebra, geometry), featuring complex content and lacking a standardized format (except for LaTeX). To reconstruct the MATH dataset, we initially filter the original data, followed by the automatic generation of new data using an LLM. Finally, the data undergo thorough manual review and proofreading.

3.1 Rules for Dataset Creation

In this section, we detail the rewriting rules for backward reasoning. For an input problem, we first split it into a set of conditions $X = \{x_1, x_2, ..., x_n\}$ and y denotes the answer. When reformulating a problem, we swap y with one of the conditions in the set X, denoted as x_k. Assuming x_k is the condition swapped, the constructed backward reasoning problem condition set can be represented as $X^{'} = \{x_1, x_2, ..., y, ..., x_n\}$, and its answer is x_k. Therefore, the backward reasoning problem and its result can be represented by $X^{'}$ and x_k respectively.

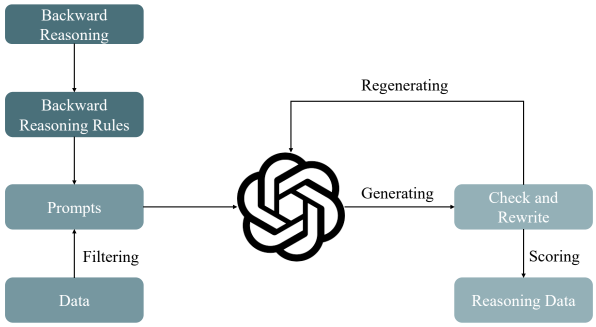

Figure 2: Backward data collection process.

3.2 Data Collection

Filtering. During the filtering phase, we conduct an initial automatical screening, eliminating cases where the question length is too short. For example, questions like "Calculate $\sqrt{2-\sqrt{2-\sqrt{2-\sqrt{2-\cdots}}}}$" which contain only one condition, cannot yield a corresponding backward reasoning problem and are therefore filtered out. Additionally, for algebra and similar questions, we conduct a meticulous manual review to ensure compliance with the rules outlined in Section 3.1.

Generating. As shown in Figure 2, the concept of backward reasoning is derived from FOBAR (Jiang et al., 2023) and has been modified and refined to develop prompts for generating backward reasoning data. We input prompts (shown in Appendix A), backward reasoning rules and data into ChatGPT to generate backward reasoning instances, which are categorized based on the types provided by MATH (Hendrycks et al., 2021), with different examples given to generate MATH backward reasoning problems in LaTeX format.

3.3 Data Review

Next, we check and rewrite the MATH problems that are able to generate backward reasoning problems but are initially generated incorrectly. We use a script to filter out cases where the answer to the

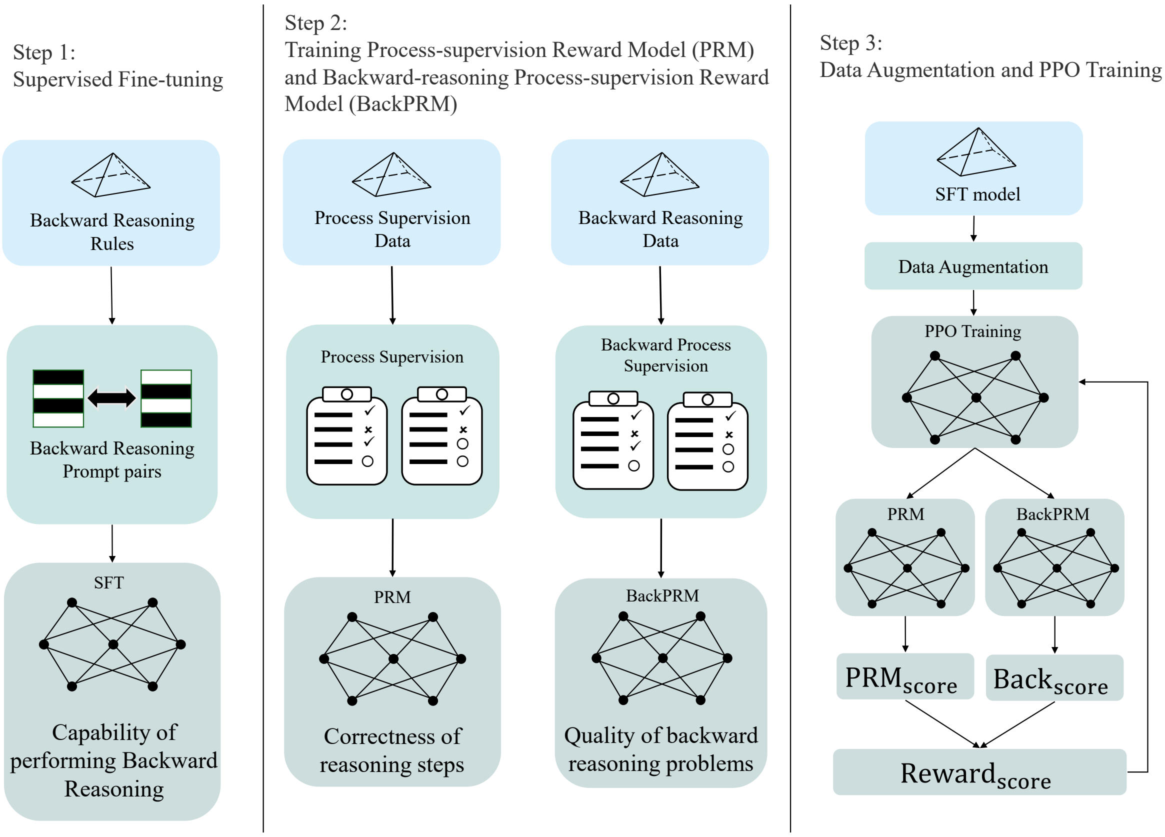

Figure 3: Diagram illustrating the three steps of our model.

backward reasoning problem is the same as that to the original problem. Most of these errors are merely semantic rephrasings of the original problem and do not adhere to the backward reasoning rule of swapping elements in y and X, described in Section 3.1. For example, the original problem "$Solve\ the\ equation: 2x+3=7, answer: x=2$" is incorrectly transformed into a backward reasoning problem "Find the value of t that satisfies $2\times t+3=7, answer: 2$". Due to ChatGPT's limited understanding of backward reasoning rules, these types of errors are the most common. Therefore, manual review and additional prompts are necessary to ensure successful problem reformulating by ChatGPT. It is particularly noteworthy that when ChatGPT is prompted so that its backward reasoning result is the same as the original problem's result (indicating an incorrect backward reasoning reformulation), it tends to directly modify the backward reasoning result to evade verification.

Finally, we input the filtered questions and reasoning steps into ChatGPT for multiple rounds of scoring the reasoning steps. Based on the scoring results, we determine the correctness of each reasoning step and average the scores from all rounds to obtain the final score for each step.

Category	**#Problems**	**#Steps**
algebra	1,713	6,202
counting & probability	770	2,334
geometry	870	2,946
intermediate algebra	1,300	4,238
number theory	860	2,228
prealgebra	1,210	3,426
precalculus	750	1,904
GSM8K	7,473	77,954
Total	14,946	101,232

Table 1: Statistics of BackMATH-Data.

3.4 Dataset Statistics

We finally collect 7.4K problems and 23K reasoning steps from MATH, and 7.4K problems and 77K reasoning steps from GSM8K. The detailed statistics of the collected dataset is shown in Table 1. Table 1 shows the number of problems and their corresponding total reasoning steps in various categories within our BackMATH-Data. In GSM8K, ChatGPT primarily uses short sentences for reasoning steps, but we divide the reasoning steps based on complete sentences, which results in a higher number of steps for GSM8K.

4 BackMATH-LLM

Inspired by InsturctGPT (Ouyang et al., 2022) and PRM (Uesato et al., 2022), we introduce the *BackMATH-LLM* training scheme in detail, which contains three stages (Supervised Fine-tuning, Reward Model training, Reinforcement Learning), as shown in Figure 3.

4.1 Supervised Fine-tuning (SFT)

Following InstructGPT (Ouyang et al., 2022), we fine-tune the model with 5K instruction-response pairs in BackMATH-Data. To enable the model to perform backward reasoning, we select pairs of original problems and their corresponding backward reasoning problems to fine-tune the model.

4.2 PRM and BackPRM

In this step, we train two reward models to supervise the quality of instructions and the correctness of each reasoning step.

PRM. This reward model is designed to assess whether each reasoning step contributes to the solution to the mathematical problem. We use 10K data from PRM800K to train the PRM for forward reasoning and rely on this PRM to evaluate the correctness of each step in the solutions generated by our model. The $\mathrm{PRM_{score}}$ is calculated as follows:

$$\mathrm{PRM_{score}} = \prod_{i=0}^{N-1} \mathrm{Step\_Score}^{i}, \quad (1)$$

where the $\mathrm{Step\_Score}^{i}$ denotes the score of each reasoning step.

BackPRM. The model is designed to assess the quality of the model's backward reasoning. We propose the BackPRM to supervise the quality of the model's backward reasoning, considering the critical role of backward reasoning in mathematical reasoning and the limited understanding of LLMs regarding backward reasoning problems. To train the BackPRM, we use 5K data from PRM800K and 5K data from our dataset, totaling 10K data instanses for training. The final reward score consists of two parts: one is the PRM score obtained by multiplying the scores of each step through process supervision, while the other is the quality score of the backward reasoning problem along with its reasoning score. The final $\mathrm{Reward_{score}}$ is calculated as follows:

$$\mathrm{Reward_{score}} = \frac{\mathrm{PRM_{score}} + \mathrm{Back_{score}}}{2}, \quad (2)$$

where the calculation method for $\mathrm{Back_{score}}$ is the same as that for the $\mathrm{PRM_{score}}$. Since forward and backward reasoning are equally important, we assign them equal weights.

4.3 Reinforcement Learning

We use the remaining 5K data from our dataset, along with GSM8K and MATH data, for Proximal policy optimization (PPO) (Schulman et al., 2017) training.

5 Experiments

This section provides an overview of our experimental setup, baseline models, and other relevant details. Subsequently, we focus on the performance metrics of our model on two popular mathematical benchmarks: GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). Our validation includes 500 samples from both the GSM8K and MATH datasets.

5.1 Experiment Settings

We fine-tuned Llama-2-7B (Touvron et al., 2023) with the data and reward models.[1] The BFLOAT16 formats and deepspeed framework were leveraged to save GPU memory and speed up training. For the SFT stages of training, we set the batch size to 4, training epoch to 3 and learning rate to 2e-5 with cosine decay. For PRM training, we used LORA technique (Hu et al., 2021) to fine-tune the lm head layer of Llama-2-7B. For PPO training, we set the learning rate to 1e-5 and the batch size to 4. All experiments were implemented in PyTorch and run on a single server with 2 NVIDIA A40 GPUs.

5.2 Baselines

We compared the performance of our model with other SOTA models, specifically WizardMath (Luo et al., 2023) and MetaMath (Yu et al., 2023), as they also enhance reasoning capabilities through data augmentation. All references of compared models are listed at Appendix G.

5.3 Main Results

As shown in Table 2, our main results indicate that BackMATH-LLM significantly outperforms other

[1]https://huggingface.co/meta-llama/Llama-2-7b-hf

Models	**GSM8K**	**MATH**
WizardMath-13B	54.9	10.7
MetaMath	66.5	19.8
GPT-3	34.0	5.2
Llama-2-7B	14.6	2.5
Llama-2-70B	56.8	13.5
Baichuan-2-7B	24.5	5.6
Baichuan-2-13B	52.8	10.1
Distilling-LM	52.3	10.0
Falcon-40B	19.6	2.9
PaLM-62B	33.0	4.4
PaLM-540B	56.5	8.8
BackMATH-LLM	**68.1**	**21.9**

Table 2: Comparison on the GSM8K and MATH datasets.

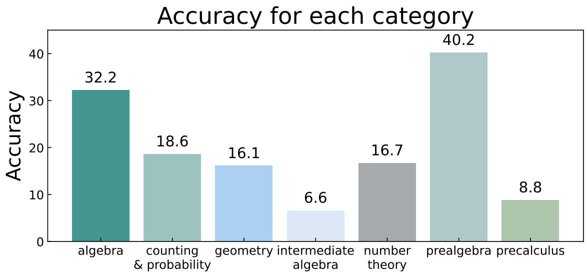

Figure 4: Detailed results on the MATH dataset.

models in mathematical problem-solving tasks. Specifically, BackMATH-LLM achieves an accuracy of 68.1% on the GSM8K dataset and 21.9% on the MATH dataset, surpassing MetaMath by 1.6% and 2.1% respectively. Compared to larger models like Llama-2-70B, BackMATH-LLM also demonstrates strong performance on both datasets. These findings highlight the substantial performance improvements of BackMATH-LLM achieved by exploring backward reasoning data.

5.4 Analysis

In this section, we provide a detailed analysis of the results on the MATH dataset, presenting the accuracy for each category, as shown in Figure 4. The model performs well on prealgebra due to their overall simplicity, making them easier to rewrite for backward reasoning. By contrast, the model struggles with intermediate algebra, as these involve more complex mathematical concepts and are more prone to errors in the reasoning steps. Appendices C, D, E and F provide more details of the case study on both datasets.

Method	Accuracy (%)
Llama-2-7B	2.5
ORM+RL	7.5
PRM+RL	12.1
SFT	6.2
SFT+ORM+RL	6.9
SFT+PRM+RL	15.1
SFT+PRM+BackPRM+RL	21.9

Table 3: Results of ablation study on the MATH dataset.

5.5 Ablation Study

In this section, we present the results of the ablation study on MATH dataset, as shown in Table 3. Specifically, our experiments are divided into two parts: one examines the effect of removing backward reasoning, and the other evaluates that of removing different modules. As the baseline model, Llama-2-7B has an accuracy of 2.5%. This result provides a benchmark for evaluating the effectiveness of other methods on MATH.

Without backward reasoning. During the SFT process, we fine-tuned the model to enable it to perform backward reasoning. Therefore, without SFT, backward reasoning is ablated, and the model only has forward reasoning capability. In the absence of backward reasoning capability, ORM+RL achieves an accuracy of 7.5%. RL with PRM feedback achieves an accuracy of 12.1%. This comparison indicates that PRM supervision is more effective than ORM supervision for the model.

Ablating modules. When the model has backward reasoning capability, i.e., after performing SFT, the accuracy of the model with only SFT is 6.2%, higher than the baseline Llama-2-7B, indicating that backward reasoning positively impacts the model's reasoning ability. SFT+ORM+RL and SFT+PRM+RL on the model achieves accuracies of 6.9% and 15.1% respectively. Among them, the result of SFT+ORM+RL is lower than ORM+RL, but SFT+PRM+RL is higher than PRM+RL. This indicates that when the model has backward reasoning capability, PRM leads to better performance. Supervised by both the PRM and the BackPRM during the reinforcement learning process, the model's accuracy reaches 21.9%. This result is significantly higher than other methods, indicating that leveraging both forward and backward reasoning data can greatly enhance the model's performance in complex reasoning tasks.

6 Conclusion

We have presented BackMATH-Data, a dataset constructed based on backward reasoning. To validate the effectiveness of BackMATH-Data in improving mathematical reasoning, we propose Backward Reasoning Process Supervision Reward Model (BackPRM) to evaluate the quality of backward reasoning problem, and BackMATH-LLM, a framework designed to enhance the backward reasoning capabilities of LLMs for solving mathematical problems. Through comprehensive experiments on the GSM8K and MATH benchmarks, we demonstrate that BackMATH-LLM significantly outperforms existing methods, achieving an accuracy of 68.1% on GSM8K and 21.9% on MATH. These findings highlight the substantial potential of backward reasoning in improving the problem-solving capabilities of LLMs.

Acknowledgements

The present research was supported by the National Key Research and Development Program of China (Grant No. 2023YFE0116400). We would like to thank the anonymous reviewers for their insightful comments.

References

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4. *arXiv e-prints*, pages arXiv–2303.

shiqi chen, Yiran Zhao, Jinghan Zhang, I-Chun Chern, Siyang Gao, Pengfei Liu, and Junxian He. 2024. FELM: Benchmarking Factuality Evaluation of Large Language Models. *Advances in Neural Information Processing Systems*, 36.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. PaLM: Scaling Language Modeling with Pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep Reinforcement Learning from Human Preferences. *Advances in neural information processing systems*, 30.

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*.

Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Supryadi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, and Deyi Xiong. 2023. Evaluating Large Language Models: A Comprehensive Survey. *arXiv preprint arXiv:2310.19736*.

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*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*.

Yufei Huang and Deyi Xiong. 2023. CBBQ: A Chinese Bias Benchmark Dataset Curated with Human-AI Collaboration for Large Language Models. *arXiv preprint arXiv:2306.16244*.

Weisen Jiang, Han Shi, Longhui Yu, Zhengying Liu, Yu Zhang, Zhenguo Li, and James Kwok. 2023. Forward-Backward Reasoning in Large Language Models for Mathematical Verification.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Alycia Lee, Brando Miranda, and Sanmi Koyejo. 2023. Beyond Scale: The Diversity Coefficient as a Data Quality Metric for Variability in Natural Language Data. *arXiv preprint arXiv:2306.13840*.

Xiaoyuan Li, Wenjie Wang, Moxin Li, Junrong Guo, Yang Zhang, and Fuli Feng. 2024. Evaluating Mathematical Reasoning of Large Language Models: A

Focus on Error Identification and Correction. *arXiv preprint arXiv:2406.00755*.

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. *arXiv 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*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On Faithfulness and Factuality in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show Your Work: Scratchpads for Intermediate Computation with Language Models. *arXiv preprint arXiv:2112.00114*.

OpenAI. 2020. Language Models are Few-Shot Learners. *arXiv preprint arXiv:2005.14165*.

OpenAI. 2022. ChatGPT: Optimizing Language Models for Dialogue.

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.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only. *arXiv preprint arXiv:2306.01116*.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling Language Models: Methods, Analysis Insights from Training Gopher. *arXiv preprint arXiv:2112.11446*.

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*.

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. *arXiv preprint arXiv:2309.15025*.

Dan Shi, Chaobin You, Jiantao Huang, Taihao Li, and Deyi Xiong. 2024. CORECODE: A Common Sense Annotated Dialogue Dataset with Benchmark Tasks for Chinese Large Language Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18952–18960.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2022. Distilling Reasoning Capabilities into Smaller Language Models. *arXiv preprint arXiv:2212.00193*.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2023. Finetuning Language Models for Factuality. *arXiv preprint arXiv:2311.08401*.

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*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in neural information processing systems*, 35:24824–24837.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2024. Fine-Grained Human Feedback Gives Better Rewards for Language Model Training. *Advances in Neural Information Processing Systems*, 36.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, JunTao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023a. Baichuan 2: Open Large-scale Language Models. *arXiv preprint arXiv:2309.10305*.

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. 2023b. Alignment for Honesty. *arXiv preprint arXiv:2312.07000*.

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*.

Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. 2023. Data-centric Artificial Intelligence: A Survey. *arXiv preprint arXiv:2303.10158*.

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*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-Tuning Language Models from Human Preferences. *arXiv preprint arXiv:1909.08593*.

A Prompts of Reformulating Problems

Here, we present an example of prompts used for ChatGPT to create backward reasoning problems. Specifically, we first provide ChatGPT with the premise for reformulating, then outline the reformulating approach, followed by reformulated examples, the questions to be reformulated, and finally the rules to be observed during the reformulation process. Table 4 shows an example of prompt in latex format.

I will give you a mathematical rule for reverse reasoning.
You need to understand it and rewrite various mathematical problems into reverse reasoning problems based on it.
I need you to rewrite the original problem into the reverse reasoning format.
You should follow: original problem: Given condition A, get result B; reverse reasoning problem: Given B, find A.
Use LaTeX's box to mark the key parts of the reverse_solution to highlight the key answer
Example: Original problem:
{Original Problem Example}
The Backward reasoning problem is:
{Backward reasoning Problem Example}
{instruction}
Note, when rewriting, pay attention to the following issues:
1. Ensure that the answer to the reverse reasoning problem is different from the answer to the original problem.
2. Avoid simple rewrites or expansions of the original problem.
3. Prevent situations where only the result of operations is given; ensure sufficient information.
4. Avoid simple verification of whether a known result meets the original problem.
5. Ensure the reverse reasoning problem and the original problem are independent.
6. New variables introduced in the original problem should not appear in the reverse reasoning problem.
7. The problem statement should not contain the required result.
8. The reverse reasoning problem should be reasoned from a different angle.
Here are my input, please output jsonl format, ensure that the answer to the reverse reasoning problem is different from the answer to the original problem:
Please use the following jsonl format to generate:
{
"reverse_problem": xxx,
"reverse_solution_process":xxx ,
"reverse_solution": xxx
}
Response:

Table 4: An example of reformualting prompts.

B Promt of Training

Table 5 shows the prompt of training our BackMATH-LLM.

```
"Below is an instruction that describes a task. "
"Write a response that appropriately completes the request."
"### Instruction:
{instruction}
let's verify step by step
### Response:"
```

Table 5: Prompt of training.

C Case Study of GSM8K

In this section, we provide a case study on GSM8K. Table 6 shows a correct case.

Instruction: Mr. Callen bought 10 paintings at $40 each and 8 wooden toys at $20 each from the crafts store to resell at a profit. However, when he sold the items, the selling price of a painting was 10% less and the selling price of a hat 15% less. Calculate the total loss Mr. Callen made from the sale of the items.
Ground Truth Answer: 64
Step1: Cost of 10 paintings at $40 each: $10 \times 40 = 400$ Step2: Cost of 8 wooden toys at $20 each: $8 \times 20 = 160$ Step3: Total cost: $400 + 160 = 560$ Step4: Selling price of each painting (10% less): $40 - (0.10 \times 40) = 40 - 4 = 36$ Step5: Total selling price for paintings: $10 \times 36 = 360$ Step6: Selling price of each wooden toy (15% less): $20 - (0.15 \times 20) = 20 - 3 = 17$ Step7: Total selling price for wooden toys: $8 \times 17 = 136$ Step8: Total selling price: $360 + 136 = 496$ Step9: Total loss: $\text{Total Cost} - \text{Total Selling Price} \quad 560 - 496 = 64$

Table 6: Case study on GSM8K.

Then, we show a case study of solving backward reasoning problem step by step in Table 7.

Instruction: Mr. Callen bought x paintings at \$40 each and 8 wooden toys at \$20 each from the crafts store to resell at a profit. However, when he sold the items, the selling price of a painting was 10% less and the selling price of a hat 15% less. Calculate the total loss Mr. Callen made from the sale of the items. If we know the answer is 64, what is the value of x?

Ground Truth Answer: 10

Step1: Cost of x paintings at \$40 each: $40x$

Step2: Cost of 8 wooden toys at \$20 each: $8 \times 20 = 160$

Step3: Total cost: $40x + 160$

Step4: Selling price of each painting (10% less): $40 - (0.10 \times 40) = 40 - 4 = 36$

Step5: Total selling price for paintings: $36x$

Step6: Selling price of each wooden toy (15% less): $20 - (0.15 \times 20) = 20 - 3 = 17$

Step7: Total selling price for wooden toys: $8 \times 17 = 136$

Step8: Total selling price: $36x + 136$

Step9: Total loss: Total Cost - Total Selling Price

Step10: $40x + 160 - (36x + 136) = 4x + 24$

Step11: $40x + 160 - 36x - 136 = 4x + 24$

Step12: $4x + 24 = 64$

Step13: $4x = 64 - 24$

Step14: $4x = 40$

Step15: $x = \frac{40}{4}$

Step16: $x = 10$

Table 7: Case study on GSM8K.

D Error Study of GSM8K

Then, we show an example of erroneous reasoning from GSM8K step by step in Table 8.

Instruction: A professional company is hiring for a new position. They have two qualified applicants. The first applicant will accept a salary of $42000 and make the company $93000 in the first year, but needs 3 months of additional training that costs $1200 a month. The second applicant does not need training and will make the company $92000 in the first year, but is requesting a salary of $45000 and a hiring bonus of 1% of his salary. Less the amount it will cost to pay for each candidate, how many more dollars will one candidate make the company than the other in the first year?

Ground Truth Answer: 850
Step1: Calculate the cost for the first applicant:
Step2: Salary: $42000
Step3: Total cost: $42000
Step4: Calculate the net profit for the first applicant
Step5: Revenue: $93000
Step6: Total cost: $42000
Step7: Net profit: 93000 - 42000 = $51000
Step8: Calculate the cost for the second applicant
Step9: Salary: $45000
Step10: Total cost: $45000
Step11: Calculate the net profit for the second applicant
Step12: Revenue: $92000
Step13: Total cost: $45000
Step14: Net profit: 92000 - 45000 = $47000
Step15: Calculate the difference in net profit between the two applicants
Step16: Net profit for the first applicant: $51000
Step17: Net profit for the second applicant: $47000
Step18: Difference in net profit: 51000 - 47000 = $4000

Table 8: An example with erroneous reasoning from GSM8K.

E Case Study of MATH

Instruction: Given that α and β are the roots of $x^2 - 2x - 1 = 0$, find $5\alpha^4 + 12\beta^3$.

Ground Truth Answer: 169

Step1: The given quadratic equation is:

$$x^2 - 2x - 1 = 0$$

using the quadratic formula $x = \frac{-b\pm\sqrt{b^2-4ac}}{2a}$, where $a = 1$, $b = -2$, and $c = -1$, we get:

$$x = \frac{2 \pm \sqrt{(-2)^2 - 4 \cdot 1 \cdot (-1)}}{2 \cdot 1} = \frac{2 \pm \sqrt{4+4}}{2} = \frac{2 \pm \sqrt{8}}{2} = \frac{2 \pm 2\sqrt{2}}{2} = 1 \pm \sqrt{2}$$

thus, the roots are:

$$\alpha = 1 + \sqrt{2} \quad \text{and} \quad \beta = 1 - \sqrt{2}$$

Step2: Calculate α^4 and β^3, first, we find α^2:

$$\alpha^2 = (1 + \sqrt{2})^2 = 1 + 2\sqrt{2} + 2 = 3 + 2\sqrt{2}$$

next, we find α^4:

$$\alpha^4 = (\alpha^2)^2 = (3 + 2\sqrt{2})^2 = 9 + 12\sqrt{2} + 8 = 17 + 12\sqrt{2}$$

.

Step3: Calculate β^3 First, we find β^2:

$$\beta^2 = (1 - \sqrt{2})^2 = 1 - 2\sqrt{2} + 2 = 3 - 2\sqrt{2}$$

next, we find β^3:

$$\beta^3 = \beta \cdot \beta^2 = (1 - \sqrt{2})(3 - 2\sqrt{2}) = 3 - 2\sqrt{2} - 3\sqrt{2} + 4 = 7 - 5\sqrt{2}$$

Step4: Calculate $5\alpha^4 + 12\beta^3$ Now, we substitute the values of α^4 and β^3 into the expression:

$$5\alpha^4 + 12\beta^3 = 5(17 + 12\sqrt{2}) + 12(7 - 5\sqrt{2})$$

Distribute the constants:

$$= 85 + 60\sqrt{2} + 84 - 60\sqrt{2}$$

Combine like terms:

$$= 85 + 84 = 169$$

Thus, the value of $5\alpha^4 + 12\beta^3$ is:

$$\boxed{169}$$

Table 9: Case study on MATH.

Instruction: Given $malpha^4 + 12beta^3 = 169$ where $alpha, beta$ are roots of $x^2 - 2x - 1 = 0$, find m.

Ground Truth Answer: 5

Step 1: Solve the Quadratic Equation
The quadratic equation is:

$$x^2 - 2x - 1 = 0$$

Solving for the roots, we get:

$$\alpha = 1 + \sqrt{2}, \quad \beta = 1 - \sqrt{2}$$

Step 2: Calculate α^4 and β^3
First, we find:

$$\alpha^2 = (1 + \sqrt{2})^2 = 3 + 2\sqrt{2}$$

$$\alpha^4 = (3 + 2\sqrt{2})^2 = 17 + 12\sqrt{2}$$

Similarly,

$$\beta^2 = (1 - \sqrt{2})^2 = 3 - 2\sqrt{2}$$

$$\beta^3 = (1 - \sqrt{2})(3 - 2\sqrt{2}) = 7 - 5\sqrt{2}$$

Step 3: Substitute α^4 and β^3 into the Given Expression We have:

$$m(17 + 12\sqrt{2}) + 12(7 - 5\sqrt{2}) = 169$$

Combine terms:

$$17m + 84 + 12m\sqrt{2} - 60\sqrt{2} = 169$$

Separate the rational and irrational parts:

$$17m + 84 = 169 \quad \text{and} \quad 12m\sqrt{2} - 60\sqrt{2} = 0$$

Step 4: Solve for m From the rational part:

$$17m + 84 = 169 \quad \Rightarrow \quad 17m = 85 \quad \Rightarrow \quad m = 5$$

Thus, the value of m is:

$$\boxed{5}$$

Table 10: Case Study on MATH.

F Error Study of MATH

Instruction: The graph of $x^2 - 2xy + 3y^2 = 5$ is an ellipse, although its axes are not parallel to the coordinate axes. Two horizontal lines and two vertical lines lie tangent to the ellipse, forming a rectangle, as shown: [asy] (7cm); (rotate(20)*xscale(2.4)*unitcircle); ((-3.5,0)–(3.5,0),EndArrow); ((0,-2.5)–(0,2.5),EndArrow); r1=2.29; draw((r1,-2.2)–(r1,2.2),dotted); ((-r1,-2.2)–(-r1,2.2),dotted); r2=1.26; draw((-3,r2)–(3,r2),dotted); ((-3,-r2)–(3,-r2),dotted); (x,(3.5,0),E); label(y,(0,2.5),N);[/asy]What is the area of the rectangle?"

Ground Truth Answer: $10\sqrt{3}$

Step1: Solve the quadratic equation:

$$x^2 - 2x - 1 = 0$$

Using the quadratic formula, we find the roots:

$$\alpha = 1 + \sqrt{2}, \quad \beta = 1 - \sqrt{2}$$

Step2: Determine the rotation angle: Using the formula for the angle of rotation to eliminate the xy term:

$$\tan 2\theta = \frac{B}{A - C}$$

where $A = 1$, $B = -2$, and $C = 3$:

$$\tan 2\theta = \frac{-2}{1-3} = \frac{-2}{-2} = 1 \Rightarrow 2\theta = \frac{\pi}{4} \Rightarrow \theta = \frac{\pi}{8}$$

Step3: Rotate the coordinate system: Using $\theta = \frac{\pi}{8}$, the rotation matrix elements are:

$$x = x' \cos\frac{\pi}{8} - y' \sin\frac{\pi}{8}, \quad y = x' \sin\frac{\pi}{8} + y' \cos\frac{\pi}{8}$$

Substitute these into the original equation $x^2 - 2xy + 3y^2 = 5$.
Step4: Simplify to the standard ellipse equation: After substituting and simplifying, we get:

$$A'x'^2 + C'y'^2 = 5$$

This represents the ellipse in the rotated coordinates.
Step5: Identify the lengths of the semi-major and semi-minor axes: In the simplified equation, the coefficients A' and C' correspond to:

$$a' = \sqrt{\frac{5}{A'}}, \quad b' = \sqrt{\frac{5}{C'}}$$

Step6: Calculate the area of the rectangle: The rectangle formed by the tangents to the ellipse has sides equal to $2a'$ and $2b'$:

$$A = 2a' \times 2b' = 4 \times a' \times b'$$

Given the ellipse $x^2 - 2xy + 3y^2 = 5$, we verify the simplified form and dimensions.
Thus, the area of the rectangle is:

$$\boxed{20}$$

Table 11: An example with erroneous reasoning from MATH.

G References of Compared Models

We list the compared models and their corresponding references in Table 12.

Model	Reference
WizardMath-13B	Luo et al. (2023)
MetaMath	Yu et al. (2023)
GPT-3	OpenAI (2020)
Llama-2-7B Llama-2-70B	Touvron et al. (2023)
Baichuan-2-7B Baichuan-2-13B	Yang et al. (2023a)
Distilling-LM	Shridhar et al. (2022)
Falcon-40B	Penedo et al. (2023)
PaLM-62B PaLM-540B	Chowdhery et al. (2023)

Table 12: References of Compared Models.