EnrichMath: Enriching Idea and Solution Elicit Mathematical Reasoning in Large Language Models

Anonymous ACL submission

Abstract

Large language models (LLMs) have witnessed remarkable advancements across a spectrum of language tasks. Despite great progress, mathematical problem-solving is still a particularly formidable challenge. Previous studies have tried to address this problem by augmenting questions and found that the performance is saturated with more training data. To further enhance the complex mathematical reasoning capabilities of LLMs, we propose EnrichMath, which is fine-tuned on our EnrichMathQA dataset. EnrichMathQA is constructed by enhancing answers in MATH and GSM8K to have a leading summary and reducing the thought jumping with our proposed Enrich Reasoning Idea(ERI) and Enrich Reasoning Solution(ERS) strategies. EnrichMath achieves state-of-the-art performance among current open-source mathematical models. Our EnrichMath-70B achieves 32.5% accuracy on the MATH benchmark, outperforming MetaMath by 2.7%. Furthermore, EnrichMath-70B achieves an accuracy of 84.1% on GSM8K, which is comparable to the methods that use external calculation tools.

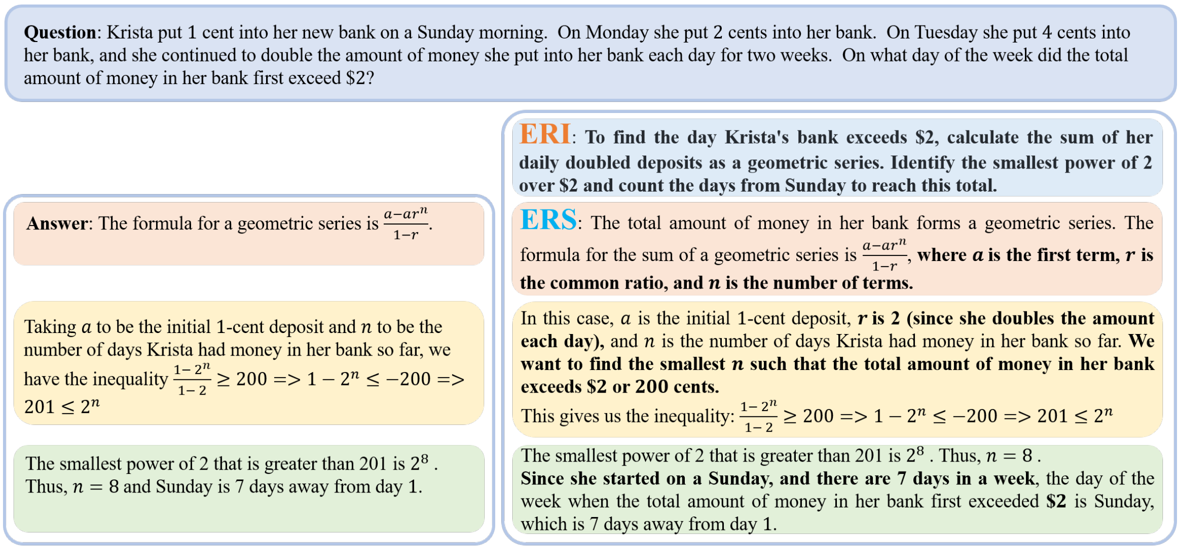
1 Introduction

Recently, large language models (LLMs) have demonstrated significant advancements among various NLP tasks, such as coding assistance(Chen et al., 2021; Li et al., 2023), adherence to instructions(Brown et al., 2020; Ouyang et al., 2022), and mathematical problem-solving(Collins et al., 2023; Imani et al., 2023; Lu et al., 2022). Mathematical problem-solving is particularly challenging for current LLMs, as it requires precise multi-step reasoning capabilities. The final answer to a mathematical problem highly relies on the accurate output of every previous step, which is very difficult for current LLMs to produce. At present, the research community is making efforts to increase the mathematical problem-solving ability of open-source LLMs (such as LLaMA(Touvron et al., 2023a)) since the training recipes of high-ranking models (such as GPT-4(Achiam et al., 2023)) are still close-sourced.

To boost the mathematical problem-solving power of open-source LLMs, researchers have mainly developed two strategies: prompt-based and fine-tuning approaches. Prompt-based strategies(Fu et al., 2022; Wei et al., 2022) aim to utilize the model's inherent reasoning abilities through in-context learning without modifying the model's parameters, such as crafting special prompts and selecting illustrative examples. In contrast, fine-tuning methods(Xu et al., 2023; Longhui Yu, 2023; Chiang et al.; Yuan et al., 2023; Gou et al., 2023; Wang et al., 2023a) finetune the model parameters with the output from other well-performed closed-source LLMs. Due to the prompt-based strategy needing to find different prompts for different LLMs which is less effective, recent studies are mainly focused on the finetuning-based strategy.

Training data plays a vital role in finetuning LLMs. Recently proposed MetaMath(Longhui Yu, 2023) generates large-scale training data by augmenting mathematical questions with forward and backward reasoning directions. By finetuning LLaMA-2 on their collected new training data, the finetuned model MetaMath achieves state-of-the-art performance on two popular benchmarks. However, the MetaMath approach mainly focuses on augmenting questions, which could only bootstrap question understanding ability. The model still suffers from hallucinations in the answers such as calculating and logical errors.

To enhance the answering ability of LLMs, we introduce two novel methodologies inspired by the process of human practice: Enrich Reasoning Idea (ERI) and Enrich Reasoning Solution (ERS). As illustrated in Fig.1, the ERI sketches the overall solution at the beginning of the answer, which serves as a strategic guide to the following rea-



Question: Krista put 1 cent into her new bank on a Sunday morning. On Monday she put 2 cents into her bank. On Tuesday she put 4 cents into her bank, and she continued to double the amount of money she put into her bank each day for two weeks. On what day of the week did the total amount of money in her bank first exceed $2?

Answer: The formula for a geometric series is $\frac{a-ar^n}{1-r}$.

Taking a to be the initial 1-cent deposit and n to be the number of days Krista had money in her bank so far, we have the inequality $\frac{1-2^n}{1-2} \geq 200 => 1-2^n \leq -200 => 201 \leq 2^n$

The smallest power of 2 that is greater than 201 is 2^8 . Thus, $n = 8$ and Sunday is 7 days away from day 1.

original answer

ERI: **To find the day Krista's bank exceeds $2, calculate the sum of her daily doubled deposits as a geometric series. Identify the smallest power of 2 over $2 and count the days from Sunday to reach this total.**

ERS: The total amount of money in her bank forms a geometric series. The formula for the sum of a geometric series is $\frac{a-ar^n}{1-r}$, **where a is the first term, r is the common ratio, and n is the number of terms.**

In this case, a is the initial 1-cent deposit, **r is 2 (since she doubles the amount each day),** and n is the number of days Krista had money in her bank so far. **We want to find the smallest n such that the total amount of money in her bank exceeds $2 or 200 cents.**
This gives us the inequality: $\frac{1-2^n}{1-2} \geq 200 => 1-2^n \leq -200 => 201 \leq 2^n$

The smallest power of 2 that is greater than 201 is 2^8 . Thus, $n = 8$.
Since she started on a Sunday, and there are 7 days in a week, the day of the week when the total amount of money in her bank first exceeded **$2** is Sunday, which is 7 days away from day 1.

EnrichMath answer

Figure 1: Overview of the mathematical problem-solving LLM - EnrichMath.(The areas in bold are optimizations compared to the original answer.)

soning process. The ERS, on the other hand, aims to reduce the "thought jumps" that lead to abrupt and illogical transitions between steps by adding more logical transitions and missing context. It is worth noting that, our proposed ESR strictly follows the original answer without involving more mathematical calculations or symbols. This could generate highly accurate enhanced answers that would significantly reduce the cost of data collection. Combining these two strategies, we enhance the reasoning flow of current answers on two popular mathematical datasets (MATH(Hendrycks et al., 2021) and GSM8K(Cobbe et al., 2021)) to generate a new EnrichMathQA dataset. We finetune the open-source LLaMA-2 model on the generated EnrichMathQA dataset and name the new model EnrichMath. By summarizing the ideas of solving the problem and filling the logical gap and missing contexts in solutions, our proposed EnrichMath model achieves state-of-the-art accuracy on both datasets.

Another benefit of our method is that our strategy is orthometric to previous question-based methods. The recently proposed method MetaMath achieves promising performance by augmenting questions. However, they found that combining with previously proposed mathematical reasoning data results in worse performance under data sizes from 20k to 100k, concluding that "more data is not always better". Our method aims to improve the answering ability of LLMs, which is complementary to previous methods that aim to understand the questions better. Our extended experiments indicate that combining MetaMathQA with our EnrichMathQA could bring further performance improvements, and more data performs better. We encourage the community to continue investigating effective data augmentation methods for mathematical problem-solving.

Our contributions can be summarized below:

- We propose two strategies to enrich the answers in the training data to generate EnrichMathQA: Enrich Reasoning Idea (ERI) to summarize the idea at the beginning and Enrich Reasoning Solution (ERS) to address the thought jumping in answers.

- By fine-tuning the open-source LLaMA-2 model on our collected EnrichMathQA dataset, we obtain the EnrichMath model, which has an excellent performance in solving mathematical problems.

- Our methods focus on improving the answering ability, which is orthometric to previous question augmentation methods. Extensive experiments indicate that more data is better based on our strategies.

- Our proposed EnrichMath model significantly surpasses current state-of-the-art methods on two popular mathematical benchmarks, MATH and GSM8K. Specifically, EnrichMath attains an accuracy of 32.5% on the

MATH dataset and 84.1% on GSM8K, marking an improvement of 2.7% on MATH over MetaMath. Besides, EnrichMath is comparable to the methods using external tools on the GSM8K benchmark, even outperforming MathCoder.

2 Related Work

2.1 Large Language Model

LLMs have witnessed significant changes in various natural language processing (NLP) tasks, yielding unprecedented improvements. These models are distinguished by their extensive scale, harnessing tens to hundreds of billions, and even trillions, of parameters, trained on vast datasets. LLMs have not only demonstrated exceptional performance across a wide range of downstream tasks but have also shown emergent capabilities that were previously unattainable(Zhao et al., 2023).

LLMs can be divided into two categories: closed-source and open-source. Among the closed-source models, notable examples include OpenAI's GPT-4, Google's LaMDA(Thoppilan et al., 2022), PaLM(Chowdhery et al., 2023), Bard(Manyika and Hsiao, 2023), and DeepMind's Chinchilla(Hoffmann et al., 2022) and Gopher(Rae et al., 2021). On the other hand, the open-source domain has witnessed a flourishing development of LLMs. EleutherAI contributes GPT-NeoX-20B(Black et al., 2022) and GPT-J-6B(Wang and Komatsuzaki, 2021). BigScience introduces BLOOM(Workshop et al., 2022), whose models range from 7B to 176B parameters. The most popular model in the open-source LLM community comes from Meta's LLaMA-1(Touvron et al., 2023a) and LLaMA-2(Touvron et al., 2023b). These models are available in three parameter configurations: 7B, 13B, and 70B. Building upon the LLaMA base model, various models fine-tuned on it have emerged, including Alpaca(Taori et al., 2023), Vicuna(Chiang et al.), Guanaco(Dettmers et al., 2023), WizardLM(Xu et al., 2023), and others. LLaMA has gained popularity in the open-source community, providing a versatile foundation for model fine-tuning and application-specific tuning.

2.2 Mathematical Reasoning of Large Language Model

Mathematical reasoning is a key aspect of human intelligence that enables us to comprehend and make decisions based on numerical data and language(Lu et al., 2022). This cognitive faculty is also an important factor in evaluating the capabilities of LLMs. Mathematical reasoning is still a great challenge for LLMs, which struggle with complex computations and symbolic manipulations. To improve reasoning capabilities, prompt-based methods are proposed. Chain-of-thought prompting (CoT)(Wei et al., 2022) proposes that LLMs can improve reasoning capabilities by generating reasoning chains through leveraging intermediate natural language rationales as prompts. Some recent studies also proposed to select in-context-learning examples, since the chosen examples in prompts have a large impact on the accuracy and stability of reasoning(Rubin et al., 2021; Zhang et al., 2022).

Fine-tuning is another way to improve reasoning capabilities. WizardMath(Xu et al., 2023) introduces an evolutionary instruction method, evolving questions to varying levels of complexity to generate a spectrum of difficulty. MetaMath(Longhui Yu, 2023), on the other hand, employs a bootstrapping question approach from diverse perspectives, including rephrasing, self-verification, and FOBAR questions, thereby increasing the diversity of questions. Despite the progress in question enhancement, there remains a paucity of research dedicated to the enhancement of answers to further improve reasoning ability. One method to enhance answers is to extend the reasoning paths by LLMs, but this often results in lower accuracy and higher computational costs(Gou et al., 2023). Besides, recent methodologies have suggested the use of computational tools to improve reasoning accuracy, such as Python(Gou et al., 2023; Wang et al., 2023a), which relies on computation tools to ensure accuracy.

Different from most prior work that focused on augmenting questions, this paper targets answer augmentation, which can be cooperated with the question-enhancing methods to further improve performance. Besides, we take a more thorough study in eliciting LLMs' intrinsic reasoning capabilities to derive solutions rather than relying on external calculation tools.

3 Method

3.1 Enrich Reasoning Idea

In human problem-solving, the initial step typically involves thinking of a solution strategy, which then guides the systematic derivation of the answer. However, current mathematical datasets (like MATH and GSM8K) typically favor direct problem-solving without any pre-thinking or ideas. Inspired by this human-like approach to problem-solving, we propose the concept of Enrich Reasoning Idea (ERI), enriching the answer by constructing an idea first. In this way, when answering a question, LLMs will generate an idea first, and then generate the answer following the idea.

To construct the dataset, ERI generates a succinct yet logically robust idea derived from the given answer using good-performance closed-source LLMs GPT-4, and the prompt can be found in the Appendices. One specific example is illustrated in Example 3.1. ERI pointed out the problem-solving trajectory: it begins by applying the volume formula to get the radius, followed by employing the surface area formula to compute the surface area.

Furthermore, ERI can be seamlessly added to the front of the original answer. This integration serves to direct the inferential problem-solving trajectory. By adhering to this structured premise, LLMs are compelled to engage in a stepwise deductive reasoning process following the ideas, thereby mitigating the risk of deviation from the intended analytical pathway. The incorporation of ERI augments the resolution process, yielding a more comprehensive and coherent solution narrative.

3.2 Enrich Reasoning Solution

Although LLMs have achieved great progress in capturing and utilizing knowledge information, they still suffer from hallucinations(Zhao et al., 2023). In mathematical problem-solving, these hallucinations may occur as incorrect solutions or reasoning steps that lack a logical basis, often due to the model's inability to fully grasp the context or the implicit knowledge assumed in the problem statement. We propose that a detailed contextualization of each reasoning step, with explicit logical connections, can mitigate the occurrence of hallucinations to enhance the mathematical reasoning capabilities of LLMs. However, many current mathematical datasets exhibit "thought jumps", omitting contexts like some common sense, meanings of formula variables, causal relationships, etc.

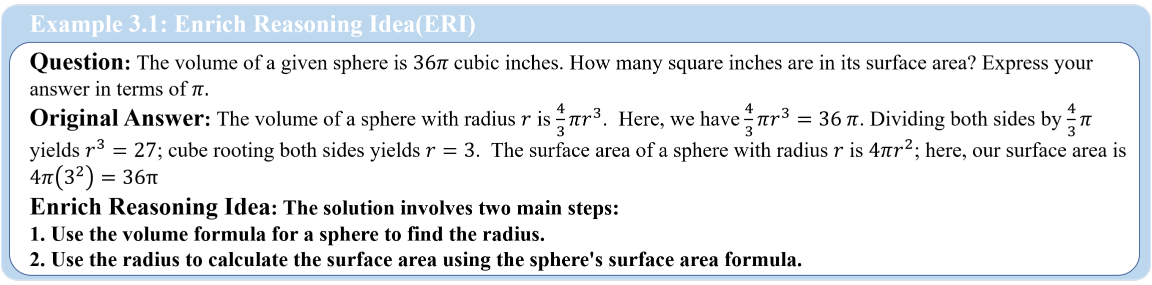
To mitigate this issue, we introduce the Enrich Reasoning Solution (ERS) strategy, specifically designed to avoid "thought jumps". ERS bridges the gaps in contextual information that often result in non-sequitur steps within problem-solving sequences. ERS will not modify the solution pathway, formulas, or calculations in the original answer. Instead, ERS aims to supplement the missing context and logical gap in the reasoning process, thereby elucidating the rationale behind each step of the model's reasoning process. Equipping ERS, the model gains a deeper insight into the internal logic and structural relationships, which helps to bolster the reasoning ability.

An example is illustrated in Example 3.2. Compared to the original answer, ERS explains in detail what the combination formula means and why it should be used. Furthermore, ERS points out the independence of choosing yogurt flavors and choosing toppings, which explains why the number of choices for the two can be multiplied to get the total number of combinations.

Most previous augmenting methods on mathematical datasets discard the original answers and use LLMs to generate their expected answers. Some request LLMs to give more steps or new reasoning paths. Such data augmentation methods have high requirements for the reasoning ability and accuracy of LLMs. However, GPT-4, which is currently recognized as having strong reasoning capabilities, can only achieve an accuracy of 42.5% on the MATH test set, so half of the data generated in this way may be wrong. This process is resource-intensive and necessitates the reacquisition of erroneous samples.

In contrast, given the original questions and answers, ERS employs enriching instruction and few-shot prompting to guide LLMs in enriching the original solutions. Compared to other accuracy-low methods, ERS achieves an accuracy of about 99.7% when constructing enriched solutions. The key to achieving high accuracy is that ERS does not require LLMs to generate new reasoning pathways, but only enriches existing reasoning steps, which reduces the requirements for LLMs. The prompt can be found in Appendices.

Based on the assumption that "thought jumps" in reasoning lead to hallucinations in the model, ERS mandates that the enrichment process must avoid any form of thought jumps, thereby ensuring a seamless and logically coherent solution narrative. Therefore, the enriched answers are not only more rigorous but also exhibit a smooth, step-by-step explanation that is contextually grounded.



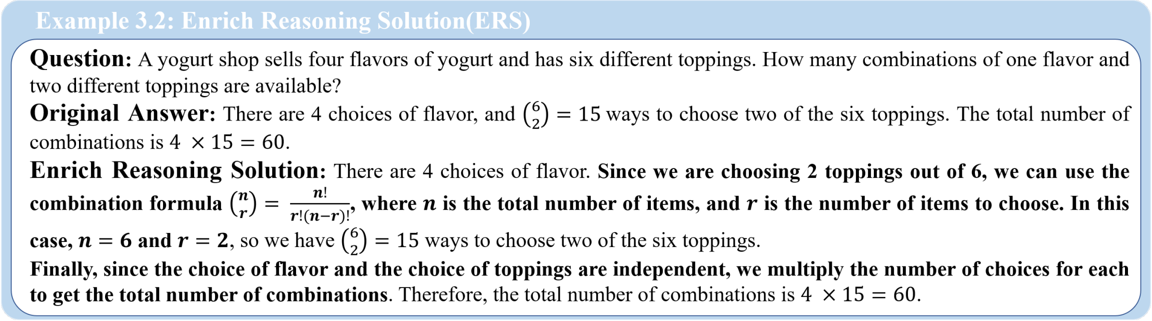
Example 3.1: Enrich Reasoning Idea(ERI)

Question: The volume of a given sphere is 36π cubic inches. How many square inches are in its surface area? Express your answer in terms of π.

Original Answer: The volume of a sphere with radius r is $\frac{4}{3}\pi r^3$. Here, we have $\frac{4}{3}\pi r^3 = 36\,\pi$. Dividing both sides by $\frac{4}{3}\pi$ yields $r^3 = 27$; cube rooting both sides yields $r = 3$. The surface area of a sphere with radius r is $4\pi r^2$; here, our surface area is $4\pi(3^2) = 36\pi$

Enrich Reasoning Idea: The solution involves two main steps:
1. Use the volume formula for a sphere to find the radius.
2. Use the radius to calculate the surface area using the sphere's surface area formula.

Figure 2: Example of Enrich Reasoning Idea



Example 3.2: Enrich Reasoning Solution(ERS)

Question: A yogurt shop sells four flavors of yogurt and has six different toppings. How many combinations of one flavor and two different toppings are available?

Original Answer: There are 4 choices of flavor, and $\binom{6}{2} = 15$ ways to choose two of the six toppings. The total number of combinations is $4 \times 15 = 60$.

Enrich Reasoning Solution: There are 4 choices of flavor. **Since we are choosing 2 toppings out of 6, we can use the combination formula $\binom{n}{r} = \frac{n!}{r!(n-r)!}$, where n is the total number of items, and r is the number of items to choose. In this case, $n = 6$ and $r = 2$**, so we have $\binom{6}{2} = 15$ ways to choose two of the six toppings.
Finally, since the choice of flavor and the choice of toppings are independent, we multiply the number of choices for each to get the total number of combinations. Therefore, the total number of combinations is $4 \times 15 = 60$.

Figure 3: Example of Enrich Reasoning Solution

4 Experiments

4.1 Datasets

We propose EnrichMathQA and use it as the training dataset. Besides, we evaluate our model on two popular benchmarks: MATH and GSM8K.

MATH This dataset consists of competition-level mathematics problems. It encompasses a total of 12,500 problems, partitioned into 7,500 for training and 5,000 for testing. Each problem is accompanied by a step-by-step solution and concludes with a distinct final answer, which is formatted for straightforward verification of the model-generated solutions. Notably, the MATH dataset spans a broad spectrum of subjects and difficulty levels, including seven categories: Prealgebra, Algebra, Number Theory, Counting and Probability, Geometry, Intermediate Algebra, and Precalculus.

GSM8K Comprising a diverse collection of grade school mathematical word problems, GSM8K is recognized for its high quality. While it is generally considered less challenging than the MATH dataset, it similarly provides step-level solutions with basic arithmetic operations (addition, subtraction, multiplication, division). The GSM8K dataset contains 8,500 problems, with 7,500 for training and 1,000 for testing.

EnrichMathQA Table 1 illustrates the composition of our novel EnrichMathQA dataset, which is constructed by LLM with different temperatures. This dataset is enriched based on MATH and GSM8K, containing 70K pairs of high-quality problems and enriched answers(35K from EnrichMathQA-MATH and 35K from EnrichMathQA-GSM8K). Utilizing this comprehensive EnrichMathQA dataset, we proceeded to conduct a series of experiments on both the MATH and GSM8K benchmarks

Table 1: Number of samples in our proposed EnrichMATHQA.

Datasets	ERI	ERS
EnrichMATHQA-MATH	35K	35K
EnrichMATHQA-GSM8K	35K	35K

4.2 Models and Experimental Details

We use the most popular and state-of-the-art open-source large language model LLaMA-2 as the base model for finetuning. GPT-4 is used to construct MetaMathQA.

Following (Longhui Yu, 2023; Xu et al., 2023), we use the AdamW optimizer to train the model with 3 epochs and the learning rate is set as 2e-5. The batch size is 32 for the 70B model and 128 for the 13B model. The ablation study experiments in Table 2 use the same data size of MATH and GSM8K to get a fair comparison. The experiments on analysis of more data for finetuning are also

Example 6.1: Case Study in Test Set of MATH

Question: The product $ab = 1200$, a is an integer, and b is an odd integer. What is the largest possible value of b?

SFT Answer: Since b is odd, it can be written as $2k+1$, where k is an integer. Thus, $ab = a(2k+1) = 2ak + a = 1200$. As a is an integer, the largest possible value of b is when $a = 1$, in which case $b = 1200$ (and $k = 599$). If b is any larger, then a becomes a fraction, so $b = 1200$ is the largest possible value. (✗)

MetaMath Answer: Since a is an integer, b must be a factor of 1200.
The prime factorization of 1200 is $2^3 \times 3 \times 5^2$.
To maximize b, we want to choose the largest power of 5, which is $5^2 = 25$.
Therefore, the largest possible value of b is 25.
The answer is: 25. (✗)

Our Answer: **(ERI)** **To find the largest possible value of b, consider the prime factorization of 1200 and the fact that b must be odd. Since b cannot include the factor of 2 present in 1200, the largest odd factor of 1200 that can be b is the product of the odd prime factors raised to the appropriate power.**

(ERS) To solve this, we first express 1200 as a product of its prime factors:

$$1200 = 2^3 \times 3 \times 5^2$$

Since b is an odd integer, it cannot include the factor of 2 that is present in the prime factorization of 1200. Therefore, b must be composed only of the odd prime factors of 1200, which are 3 and 5. To maximize b, we should use the highest powers of these primes that are factors of 1200.

The highest power of 3 that divides 1200 is 3^1, and the highest power of 5 is 5^2. Multiplying these together gives us the largest possible value for b :

$$b = 3^1 \times 5^2 = 3 \times 25 = 75$$

Therefore, the largest possible value of b is 75.
The answer is: 75. (✓)

Figure 4: Example of Case Study in Test Set of MATH

Table 2: Ablation study of ERI, ERS, and MetaMath on MATH and GSM8K dataset.

Datasets	ERI	ERS	MetaMath	Acc(%)
MATH				13.5
	✓			17.2
		✓		19.5
			✓	19.0
	✓	✓		21.3
	✓	✓	✓	22.6
GSM8K				56.8
	✓			72.0
		✓		76.3
			✓	74.2
	✓	✓		78.8
	✓	✓	✓	81.1

based on LLaMA-2-70B and we finally combine question bootstrapping and our answers enrichment methodologies to get the best performance.

5 Ablation Study

To evaluate the effectiveness of our new proposed Enrich Reasoning Idea(ERI) and Enrich Reasoning Solution(ERS), we conduct the ablation study on MATH and GSM8K datasets based on LLaMA-2-70B. To make a fair comparison, the training data from EnrichMathQA-MATH and EnrichMathQA-GSM8K respectively is the same size data as the training data of MATH and GSM8K, rather than repeatedly sampling to obtain more data. Furthermore, since our method focuses on enriching answers, we combine EnrichMath with MetaMath, which focuses on augmenting questions, to explore whether the combination of them could create new sparks. We also sample MetaMath with the same size as MATH and GSM8K.

5.1 Effects of ERI

As depicted in Table 2, the baseline model's accuracy is 13.5% on the MATH dataset and 56.8% on the GSM8K dataset. The ablation of ERI is conducted by integrating with ERI and the original answer, which yields an encouraging increase in accuracy to 17.2% on MATH and 72.0% on GSM8K, with an improvement of 3.7% and 15.2% respectively. It can be seen from the above results that ERI can significantly enhance the model's reasoning ability.

5.2 Effects of ERS

Table 2 depicted the performance of ERS. On the MATH dataset, the accuracy increases from 13.5% to 19.5%, achieving a 6% improvement. On GSM8K, ERS also propels the model's accuracy from 56.8% to 76.3%, obtaining a 19.5% improvement. Such huge progress is attributed to the improvement of the answers by the ERS module, which bridges the gaps in contextual information to improve the model's reasoning ability. It is worth noting that the improvement of GSM8K is greater than that of MATH, with an increase of nearly 20%, which shows that ERS improves simpler math problems more significantly(Questions from GSM8K are easier than MATH).

Finally, when integrated with ERI and ERS, our EnrichMath obtains a best performance of 21.3% on MATH and 78.8% on GSM8K. Such significant progress demonstrates the positive influence of ERI and ERS on the model's mathematical reasoning capabilities and overall accuracy.

5.3 Effects of Using MetaMath as Complement for Question Augmentation

As we know MetaMath focuses on bootstrapping questions, while EnrichMath focuses on enriching answers, so these two methods are complementary. Therefore, we conducted ablation experiments on MetaMath based on EnrichMath to improve further. To make a fair comparison, we sampled the augmented MATH data from MetaMathQA with the same size as MATH. GSM8K also performed the same operation. As shown in Table 2, by combining question and answer augmenting methodologies, the model has shown significant improvement. It achieves 1.3% and 2.3% improvement over the model only equips answer enriching on MATH and GSM8K respectively. At the same time, the performance is even more significant on the model equip question bootstrapping, with 3.6% and 6.9% improvements on MATH and GSM8K respectively.

The experiments demonstrate that augmenting questions and answers can both bolster the model's reasoning abilities, and their cooperation can further improve the accuracy of the model. Besides, with the same amount of data, augmented answers are more effective than augmented questions.

6 Comparison with Prior Works

6.1 Results on MATH and GSM8K

As is shown in Table 3, on MATH benchmark, comparing with the same open-source large language model without using an external tool(like python), our EnrichMath-70B obtains the best accuracy. EnrichMath-70B achieves 32.5% accuracy, with 2.7% improvement compared with MetaMath-70B and more than twice the accuracy of LLaMA-2-70B. For the 13B model, our EnrichMath also achieves the best performance among models of the same type achieving 23% accuracy. From the above performance, we can see the superiority of our ERI and ERS methods compared to other methods.

Table 3: Comparison of testing accuracy to existing LLMs on GSM8K and MATH test set.† means that external tools are used. ‡ means that the MetaMath is finetuned by QLoRA with the batch of 128 from (Longhui Yu, 2023), while the full finetuned version is from (Wang et al., 2023b).

Methods	Size	MATH	GSM8K
LLaMA-1	13B	3.9	17.8
LLaMA-2	13B	3.9	28.7
MPT	30B	3.1	15.2
Falcon	40B	2.5	19.6
Vicuna	13B	-	27.6
WizardMath	13B	14.0	63.9
MetaMath	13B	22.4	72.3
EnrichMath	13B	**23.0**	**73.1**
LLaMA-1	65B	10.6	50.9
LLaMA-2	70B	13.5	56.8
Platypus-2	70B	15	45.9
RFT	70B	-	64.8
WizardMath	70B	22.7	81.6
MetaMath‡	70B	26.6	82.3
MetaMath	70B	29.8	80.4
EnrichMath	70B	**32.5**	**84.1**
PAL(LLaMA-2)†	70B	18.3	55.2
MathCoder†	70B	45.1	83.9
TORA†	70B	49.7	84.3

Table 4: Comparison of testing accuracy on MATH Subtopics(70B).

MATH subtopics	WizardMath	MetaMath	EnrichMath
Intermediate Algebra	7.1	14.28	**15.0**
Precalculus	12.6	10.8	**15.4**
Geometry	15.7	20.9	**25.9**
Number Theory	16.3	23.7	**26.7**
Counting & Probability	17.3	24.3	**28.7**
Prealgebra	41.7	46.9	**50.0**
Algebra	33.3	44.56	**47.1**
Overall	22.7	29.8	**32.5**

To further analyze the improvement of our method, we show the accuracy of subtopics on MATH in Table 4. Our EnrichMath exceeds WizardMath and MetaMath among all subtopics. It's worth noting that EnrichMath achieves 15.4% on the most difficult topic, Precalculus, surpassing MetaMath 4.6% and WizardMath 2.8%.

On the GSM8K benchmark, our EnrichMath-70B achieves 84.1%, with 2.5% improvement over WizardMath and 1.8% over MetaMath with the same parameters. Furthermore, the 13B model performed even more outstandingly, surpassing WizardMath by 9.2% and MetaMath by 0.8%.

It is noteworthy that the performance of EnrichMath-70B on GSM8K is comparable to that of models utilizing external computational tools, and even outperforms MathCoder. This underscores the significant potential of using the intrinsic reasoning capabilities of LLMs without the need for additional aids.

6.2 Analysis on More Data for Fine-tuning

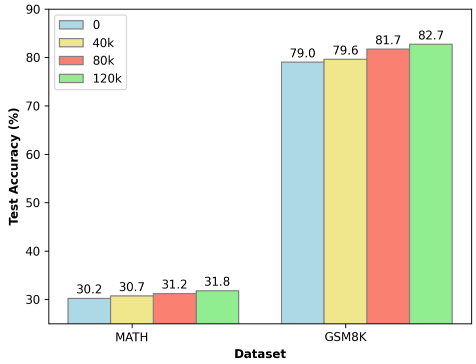


Figure 5: Performance of more data for fine-tuning on MATH and GSM8K. (Different colored bars represent different amounts sampled from MetaMathQA and combined with EnrichMathQA as the training set.)

As we know the data scaling law is significant for LLMs. MetaMath demonstrates that more data is not always better. By combining the existing augmented dataset with MetaMathQA of different scales for fine-tuning, they found that more augmented data hurt the performance. However, it's contrary to our conclusion.

As shown in Fig.5, we sample 40k, 80k, 120k from MetaMathQA to combine with EnrichMathQA for fine-tuning LLaMA-2-70B. As data scales increase, the performance of the model can be further improved, which indicates that more data is better based on our strategies.

6.3 Analysis from a Perplexity Perspective

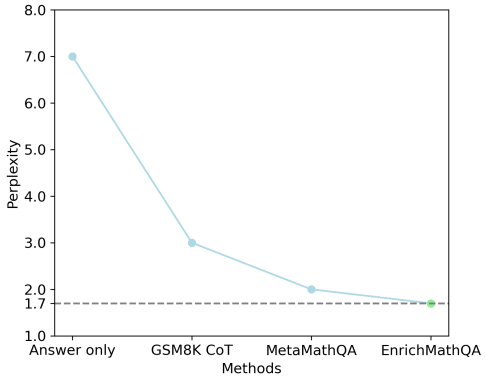


Figure 6: Perplexity of Different Method

Following (Longhui Yu, 2023), we calculate the perplexity of EnrichMathQA, which is lower than all other methods in Fig.6. The low perplexity also demonstrated that our ERI and ERS methods follow an inherently easy-to-learn nature, which facilitates eliciting the problem-solving abilities of LLMs(Longhui Yu, 2023).

6.4 Analysis on Case Study

Example 3.3 shows the case study of EnrichMath, SFT(Touvron et al., 2023b), and MetaMath on the test set of MATH. We can see that the solution from SFT is one-sided, as even if the value of a decreases, the value of b may not necessarily increase as an odd number. So ultimately b equals 1200 is wrong. As for MetaMath, the first step is true, which factors 1200 correctly. However, MetaMath fails to understand why factoring works and directly chooses 5 as the largest power, resulting in a false answer. In contrast to the above methods, firstly, our EnrichMath illustrates ERI accurately, guiding the following deviated steps to conduct prime factorization of 1200 and get the product of odd prime factors as the final answer. Secondly, EnrichMath demonstrates ERS, which performs correct factorization, and explains that the purpose of factorization is to obtain all non-even factors, including 3 and 5. As a result, EnrichMath gets the final correct answer.

7 Conclusion

In this paper, we aim to enhance the mathematical problem-solving abilities of open-source LLMs. We introduce a novel answer enhancement methodology that consists of two key components: Enrich Reasoning Idea (ERI) and Enrich Reasoning Solution (ERS), which provide a concise yet logical robust idea to guide the following reasoning process and bridge the gap of "thought jumps" by enriching existing answers. After enriching the MATH and GSM8K datasets, we got a high-quality dataset called EnrichMathQA. We then finetuned the LLaMA-2 model with our proposed EnrichMathQA dataset and got a state-of-the-art model(EnrichMath) among open-source mathematical models without using external tools. Our EnrichMath-70B achieves 32.5% on MATH and 84.1% on GSM8K, outperforming comparable LLMs by a large margin. Our work further demonstrates that with our strategy, more augmented data is better for fine-tuning, providing inspiration for data scaling law in LLMs.

Limitations

There are two limitations to this work. Firstly, our ERI and ERS methods rely on the existing answers. If the existing answer is incorrect or lacks a valid reasoning process, our proposed ERI and ERS cannot rectify it and provide a correct answer. Secondly, our EnrichMathQA's answers are longer than normal answers, which needs more resources to train and infer.

Ethics Statement

We experiment on two mathematical datasets, including GSM8K and MATH, both of which use MIT License code. The prompts used in these experiments are listed in the Appendices, and we want to emphasize that none of the prompts contain any words that discriminate against any individual or group. Furthermore, prompts would not negatively impact anyone's safety in this work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*.

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.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. 2023. *URL https://lmsys.org/blog/2023-03-30-vicuna*, 1(2):3.

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, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Katherine M Collins, Albert Q Jiang, Simon Frieder, Lionel Wong, Miri Zilka, Umang Bhatt, Thomas Lukasiewicz, Yuhuai Wu, Joshua B Tenenbaum, William Hart, et al. 2023. Evaluating language models for mathematics through interactions. *arXiv preprint arXiv:2306.01694*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*.

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*.

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*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models. *arXiv preprint arXiv:2303.05398*.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. 2023. Starcoder: may the source be with you! *arXiv preprint arXiv:2305.06161*.

Han Shi Jincheng Yu Zhengying Liu Yu Zhang James T Kwok Zhenguo Li Longhui Yu, Weisen Jiang. 2023. Metamath: Bootstrap your own mathematical questions for large language. *arXiv preprint arXiv:2309.12284*.

Pan Lu, Liang Qiu, Wenhao Yu, Sean Welleck, and Kai-Wei Chang. 2022. A survey of deep learning for mathematical reasoning. *arXiv preprint arXiv:2212.10535*.

James Manyika and Sissie Hsiao. 2023. An overview of bard: an early experiment with generative ai. *AI. Google Static Documents*, 2.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

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*.

Ben Wang and Aran Komatsuzaki. 2021. Gpt-j-6b: A 6 billion parameter autoregressive language model.

Ke Wang, Houxing Ren, Aojun Zhou, Zimu Lu, Sichun Luo, Weikang Shi, Renrui Zhang, Linqi Song, Mingjie Zhan, and Hongsheng Li. 2023a. Mathcoder: Seamless code integration in llms for enhanced mathematical reasoning. *arXiv preprint arXiv:2310.03731*.

Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Y Wu, and Zhifang Sui. 2023b. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *CoRR, abs/2312.08935*.

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.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

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

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Appendices

Prompt

Example A.1 shows the prompt of ERI. Example A.2 and Example A.3 show the prompt of both ERI and ERS for GSM8K and MATH respectively. If only ERI is needed, use Example 3.1 directly. To get both ERI and ERS, Example 3.2 or Example 3.3 can be used.

Example A.1 : Prompt for Enrich Reasoning Idea

You are a perfect assistant for answers. According to the answer I give you, I hope you give the idea of the answer to the respondent rather than the answer directly. The ideas for the answer will help the respondent to solve the question better and serve as the preparatory part of the respondent's answer.

[question]: Addison's age is three times Brenda's age. Janet is six years older than Brenda. Addison and Janet are twins. How old is Brenda?
[answer]: First, let $A = $ Addison's age, $B = $ Brenda's age, and $J = $ Janet's age. Then, from the statements given, we have the following system of equations: $$\begin{cases}
A=3B \\
J = B+6 \\
A=J
\end{cases} $$ Since $A=J$, we know that $3B=B+6$. Solving this equation, we have that $2B = 6 \Rightarrow B=3$. Thus, Brenda is $\boxed{3}$ years old.
[idea of answer]: The problem is solved by setting up a system of equations based on the relationships given in the question: Addison's age is three times Brenda's, Janet is six years older than Brenda, and Addison and Janet are twins. By equating Addison's and Janet's ages (since they are twins) and solving the resulting equation, we can find Brenda's age.

[question]: {Q}
[answer]: {A}
[idea of answer]:

Example A.2: Prompt for Enriching GSM8K(ERI and ERS)

You are an answer enricher. I hope you will refine your answers based on the questions and answers provided, avoid jumping your thinking. Pay attention to filling the missing contexts and logical gaps in existing answers. Besides, be careful not to change the original pathways or modify formulas and results, as well as special symbol expressions, such as 4 x 15 = <<4*15=60>>60. The first step is to give the idea of the answer, which helps the respondent to solve the question better. Finally, solve the problem.

[question]: Albert is wondering how much pizza he can eat in one day. He buys 2 large pizzas and 2 small pizzas. A large pizza has 16 slices and a small pizza has 8 slices. If he eats it all, how many pieces does he eat that day?
[answer]: He eats 32 from the largest pizzas because 2 x 16 = <<2*16=32>>32
He eats 16 from the small pizza because 2 x 8 = <<2*8=16>>16
He eats 48 pieces because 32 + 16 = <<32+16=48>>48
48
[enriched_answer]:
<idea of solution>To solve this, calculate the total number of slices in the large pizzas and the small pizzas separately, then add those numbers together to find the total number of slices Albert will eat.
<solution>Albert purchases two large pizzas and two small pizzas for his day of pizza eating. Each large pizza is divided into 16 slices, thus he has 2 x 16 = <<2*16=32>>32 slices from the large pizzas.
Similarly, each small pizza is divided into 8 slices. Therefore, from the small pizzas, he has 2 x 8 = <<2*8=16>>16 slices.
By adding up the slices from both the large and small pizzas, we find that Albert eats a total of 32 + 16 = <<32+16=48>>48 slices of pizza in one day.
So, if Albert consumes all the pizzas he bought, he will have eaten 48 slices of pizza in a single day.
48

[question]: {Q}
[answer]: {A}
[enriched_answer]:
<idea of solution>

Example A.3 : Prompt for Enriching MATH (ERI and ERS)

You are an answer enricher. I hope you will refine your answers based on the questions and answers provided and avoid jumping your thinking. Pay attention to filling the missing contexts and logical gaps in existing answers. Besides, be careful not to change the original pathways or modify formulas and results, as well as mathematical symbol formats. The first step is to give the idea of the answer, which helps the respondent to solve the question better. Finally, solve the problem.

[question]: Krista put 1 cent into her new bank on a Sunday morning. On Monday she put 2 cents into her bank. On Tuesday she put 4 cents into her bank, and she continued to double the amount of money she put into her bank each day for two weeks. On what day of the week did the total amount of money in her bank first exceed $\$2$?
[answer]: The formula for a geometric series is $\frac{a-ar^n}{1-r}$. Taking a to be the initial 1-cent deposit and n to be the number of days Krista had money in her bank so far, we have the inequality $$\frac{1-2^n}{1-2}\geq 200 \Rightarrow 1-2^n\leq -200 \Rightarrow 201 \leq 2^n.$$ The smallest power of 2 that is greater than 201 is 2^8. Thus, $n=8$ and $\boxed{\text{Sunday}}$ is 7 days away from day 1.
[enriched_answer]:
<idea of solution> To find the day Krista's bank exceeds $2, calculate the sum of her daily doubled deposits as a geometric series. Identify the smallest power of 2 over $2 and count the days from Sunday to reach this total.
<solution> The total amount of money in her bank forms a geometric series. The formula for the sum of a geometric series is $\frac{a-ar^n}{1-r}$, where a is the first term, r is the common ratio, and n is the number of terms.
In this case, a is the initial 1-cent deposit, r is 2 (since she doubles the amount each day), and n is the number of days Krista had money in her bank so far. We want to find the smallest n such that the total amount of money in her bank exceeds $2, or $200 cents.
This gives us the inequality:$$\frac{1-2^n}{1-2}\geq 200 \Rightarrow 1-2^n\leq -200 \Rightarrow 201 \leq 2^n.$$
The smallest power of 2 that is greater than 201 is 2^8. Thus, $n=8$.
Since she started on a Sunday, and there are 7 days in a week, the day of the week when the total amount of money in her bank first exceeded $2 is $\boxed{\text{Sunday}}$, which is 7 days away from day 1.

[question]: {Q}
[answer]: {A}
[enriched_answer]:
<idea of solution>