Exploring Reversal Mathematical Reasoning Ability for Large Language Models

Anonymous ACL submission

Abstract

Large language models (LLMs) have presented remarkable capabilities in the wide range of natural language understanding and reasoning tasks. Despite their success, a few works indicate that LLMs suffer from the "reversal curse", in which LLMs can't employ the inverted structure "B is A" when they are trained based on "A is B". To explore the effect of the "reversal curse" for LLMs on complex mathematical reasoning tasks, we present two reversal datasets upon GSM8K and MathQA and verify that LLMs also struggle to solve reversal mathematical problems. We analyze the potential reason and attribute it to the insufficient modeling of the relationship between reasoning steps caused by the left-to-right objective. Consequently, based on the characteristics of multi-step reasoning, we design a novel training method to improve the general and reversal reasoning abilities. Finally, we conduct experiments on four mathematical datasets, and the results demonstrate that our method significantly improves the general reasoning capacities and alleviates the reversal problem. *Our datasets and codes will be available at* `https://anonymous/`.

1 Introduction

With the significant increase in data and model scale, large language models (LLMs) (Brown et al., 2020; Hoffmann et al., 2022; Touvron et al., 2023; OpenAI, 2023) have emerged with their powerful multi-dimensional capabilities, such as long-context open domain conversation, code assistants (Chen et al., 2021b; Luo et al., 2023b; Wang et al., 2023; Zheng et al., 2023b), instruction following (Ouyang et al., 2022; Taori et al., 2023), particularly in the complex reasoning tasks solved by chain-of-thought (CoT) methods (Wang et al., 2022; Wei et al., 2022; Lightman et al., 2023). Nevertheless, a number of contemporary studies (Berglund et al., 2023; Grosse et al., 2023) highlight the presence of the "reversal curse" predicament in LLMs, where LLMs are trained based on the structure "A is B" in a sentence, and they cannot employ the inverted structure "B is A" to extrapolate and respond to queries effectively. Almost all works merely explore the "reversal curse" based on the name-to-description reversal task. It is worth exploring whether complex multi-step reasoning tasks also suffer from this predicament. If it exists, how should we alleviate it and improve the performance of LLMs' reasoning?

Models	GSM8K/Reversal	MathQA/Reversal
GPT-3.5-Turbo	77.4 / 52.2	63.5 / 44.6
Flan-T5-3B	13.5 / 3.5	5.8 / 5.8
Flan-T5-11B	16.1 / 12.3	15.5 / 9.6
LLama2-7B	13.7 / 7.0	19.2 / 10.3
LLama2-13B	25.3 / 10.7	25.6 / 10.3
LLama2-70B	52.1 / 30.2	42.0 / 29.7

Table 1: The accuracy of different LLMs on GSM8K, MathQA, and their correlated reversal test datasets.

To explore this problem, we choose one of the most challenging and representative reasoning tasks, i.e., mathematical problems (Collins et al., 2023; Imani et al., 2023; Luo et al., 2023a; Yuan et al., 2023), as the testbed. Resembling with the format of reversal curse problems, backward mathematical reasoning gives the answer to the original question and reverses to infer one of the variables in the question, which is first formalized by Yu et al. (2024). They construct a backward test set upon GSM8K (Cobbe et al., 2021) to evaluate the backward reasoning capabilities of LLMs. Their preliminary results on LLaMA-2-7B confirm that recent LLM can struggle to solve mathematical problems in backward rationales, leaving extensive verification and in-depth understanding unexplored.

To fill this blank, we first propose two reversal mathematical test sets to further verify that LLMs suffer from the "reversal curse" on mathematical problems. Specifically, we employ GPT-4 to imitate the format and style of original questions and

generate the reversal data based on the GSM8K and MathQA (Amini et al., 2019) test sets. The detailed construct process and data quality verification are elaborated in Section 3. After constructing two reversal test sets, we use them to evaluate representative LLMs of different model scales. As shown in Table 1, compared with original test sets, LLMs with different scales and architectures present a significant accuracy decline in the reversal datasets except for Flan-T5-3B on MathQA. This phenomenon sufficiently demonstrates that LLMs actually face difficulties in reversal reasoning in mathematical problems.

From this discovery, we analyze the potential reasons and speculate that it's related to the traditional left-to-right training objective. In the process of mathematical multi-step reasoning, LLMs strictly follow the order of deductions from left to right, which solely focuses on acquiring the association from conditions to conclusions. This shortcoming is essentially the lack of context modeling for reasoning steps, which causes the difficulty of reversal reasoning and affects the general reasoning performance. To sufficiently model the relationship of different reasoning steps, and enhance the reversal and overall reasoning ability of LLMs, we propose a simple and effective training framework, which introduces an additional bidirectional training objective based on the characteristics of multi-step deduction. In particular, we choose the partial steps as the context which employs the bidirectional attention mechanism, while utilizing the causal attention mechanism to predict the remaining unselected steps. By employing this approach, LLMs are trained to extrapolate the preceding steps in a reverse manner, drawing upon the information from the succeeding steps.

To validate the effectiveness of our method, we fine-tune Flan-T5-XL and Llama2-7B on the GSM8K dataset. Subsequently, we evaluate the performance of these models on four benchmarks and two reversal mathematical datasets. The results show that the models' general and reversal reasoning ability is superior to the latest methods that use additional training skills, and even some data augmentation strategies. Our contributions are listed below:

- We construct two reversal mathematical datasets to further explore the reversal reasoning ability of LLMs, and prove that LLMs actually suffer from the "reversal curse" on mathematical problems.

- We analyze the potential reason and attribute it to the insufficient modeling of the relationship between reasoning steps. Consequently, based on the characteristics of multi-step deduction tasks, the bidirectional training objective is designed to alleviate this problem.

- Whether on four benchmarks or two reversal datasets, applying our approach to different settings all achieves significant improvements, and even close to GPT-3.5-Turbo on the parts of benchmarks.

2 Related Work

2.1 Large Language Models

LLMs have shown impressive multi-dimensional capabilities, significantly affecting the natural language processing community (Brown et al., 2020; Hoffmann et al., 2022; Touvron et al., 2023; OpenAI, 2023). Recently, Wei et al. (2022); Wang et al. (2022) uncovered the broad prospects of CoT reasoning capabilities within LLMs. Given a few augmenting few-shot examples with multiple reasoning steps, LLMs can generate multi-step deduction toward the answer of solving complex tasks, e.g., this approach has been widely used on GPT-3.5 (OpenAI, 2022), GPT-4 (OpenAI, 2023) and LLaMA (Touvron et al., 2023) to tackle various reasoning tasks (Fu et al., 2023b; Zhang et al., 2023).

2.2 Reversal Curse

Though LLMs show impressive performance on various tasks, a number of current works (Berglund et al., 2023; Grosse et al., 2023) clarify that LLMs suffer from the reversal curse. Specifically, the autoregressive LLMs are trained on the logical sentence structure "A is B" and fail to infer "B is A". This phenomenon suggests that LLMs don't grasp the relationship of knowledge presented in the training data adequately. Lv et al. (2023) further explore this problem and contend that the reversal curse arises partly due to the specific training objectives pursued by models, mostly evident in the widespread adoption of next-token prediction techniques in causal language models. Besides, Wu et al. (2023) find that BERT is immune to the reversal curse. At the same time, a few bidirectional modeling approaches are proposed to mitigate the curse (Lv et al., 2023; Ma et al., 2023).

Prompt:
Please follow the examples that modify the question by adding the original question and answer as a new condition, only hiding one condition that appeared in the question, must keeping other conditions unchan ged, and making the hiding condition as a new question. Finally, provide the modified question and the hi ding number, and follow the for-mat as: Modified question: \n Hiding number:

Example 1:
Original question: If Ann is 9 years old and her brother is twice her age, how old will her brother be in 3 years? (The answer is 21)
Modified question: Ann's brother is twice as old as she is. In 3 years, her brother will be 27 years old. How old is Ann now?
Hiding number: 9

Example 2:
Original question: Morisette and Kael were asked to bring fruits. Morisette brought 5 apples and 8 oran ges, while Kael brought twice the amount of apples and half the number of oranges than Morisette. How many fruits do they have in total? (The answer is 27)
Modified question: Morisette and Kael were asked to bring fruits. Morisette brings 5 apples and some or anges, and Kael brings twice the apples and half the oranges that Morisette brings, they have 27 fruits in total. How many oranges does Morisette bring?
Hiding number: 8

Original question: {Q}

Figure 1: The prompt for obtaining reversal data. "Modified question" and "Hiding number" denote the generated reversal question and the corresponding answer.

2.3 Mathematical Reasoning

Mathematical multi-step reasoning, one of the most challenging problems, has attracted widespread attention. We divide the related work into two categories. One category is the prompt-based methods, another is finetuning-based. For the first one, a few approaches (Narang et al., 2023; Fu et al., 2023c; Zheng et al., 2023a; Diao et al., 2023; Li et al., 2023b) provide multiple reasoning examples to LLMs, and leverage the excellent in-context capability of LLMs to generate high-quality reasoning paths. For instance, Narang et al. (2023) entail generating various reasoning chains, potentially yielding multiple candidate answers. Among these, the answer that garners the most votes is subsequently chosen as the ultimate response. Another category is obtaining the CoT paths from closed-source LLMs (e.g., GPT-3.5, GPT-4) by employing knowledge distillation and utilizing the knowledge to fine-tune open-source models (e.g., Flan-T5, LLaMA). Yuan et al. (2023) propose the rejection sampling fine-tuning (RFT) to improve the performance through collecting more reasoning paths as augmented datasets. WizardMath (Luo et al., 2023a) applies reinforcement learning from the evol-instruct feedback method to enhance reasoning ability. MetaMath (Yu et al., 2024) adopts four data augmentation strategies to generate high-diversity data and obtain excellent performance. Li et al. (2023a) explore the effect of augmented data from multiple perspectives and put forward query and response augmentations approaches. An et al. (2023) demonstrate the effectiveness of learning from mistakes.

Besides, the potential of smaller language models (SLMs) reasoning has been verified (Magister et al., 2022; Ho et al., 2023; Fu et al., 2023a). Shridhar et al. (2023), Han et al. (2023) and Junbing et al. (2023) decompose the complex questions into a series of simpler problems. Liu et al. (2023) further distill the self-evaluation capability of LLMs into SLM to improve the performance.

3 Reversal Mathematical Datasets Construction

Different levels of mathematical word problems have been proposed to evaluate LLMs' general mathematical reasoning ability, such as

Question:
Morisette and Kael were asked to bring fruits. Morisette brought 5 apples and 8 oranges, while Kael brought twice the amount of apples and half the number of oranges than Morisette. How many fruits do they have in total? (The correct answer is 27)

CoT Reasoning:
Step 1: Morisette brought 5 apples and 8 oranges, totaling 13 fruits.
Step 2: Kael brought twice the amount of apples, so 2 * 5 = 10 apples, and half the number of oranges than Morisette, which is 8 / 2 = 4 oranges.
Step 3: Kael brought a total of 10 + 4 = 14 fruits.
Step 4: Thus, the total number of fruits they have is 13 + 14 = 27 fruits.

Figure 2: An example of the mathematical question and the corresponding CoT reasoning steps.

AddSub (Hosseini et al., 2014), MultiArith (Roy and Roth, 2015), MathQA (Amini et al., 2019), Asdiv (Miao et al., 2020), SVAMP (Patel et al., 2021), GSM8K (Cobbe et al., 2021), Math (Hendrycks et al., 2021) and so on. Moreover, Yu et al. (2024) design a reversal GSM8K dataset to evaluate the backward reasoning ability of LLMs, but the format and style of questions are different from the original, which could affect the performance of LLMs. Consequently, we construct two reversal datasets upon the GSM8K and MathQA datasets following the original format and style to further explore the reversal reasoning ability of LLMs. Specifically, we first follow SpecialFT (Fu et al., 2023a) to use 800 instances as the GSM8K test set, and artificially extract 600 medium difficulty questions from the MathQA (Amini et al., 2019). It's noticed that we change the type of MathQA from multiple-choice to answer questions. To keep the format and style of the original questions unchanged, we design a few-shot prompt (shown in Figure 1) for GPT-4 to imitate the provided examples and generate the reversal question and its corresponding answer.

Verifying the correctness of generated reversal instances is also important. To ensure the quality of generated data, we utilize GPT-4 to generate multiple reasoning results for every reversal instance and judge the consistency of results. If multiple results remain consistent, we suggest that this instance is correct. On the contrary, we manually verify the quality of the question. Specifically, If the corresponding answer could be obtained based on the description of the question, we keep it. Otherwise, we artificially construct this instance following the examples in Figure 1.

4 Methodology

In this section, we first introduce the training objective of causal mechanisms widely utilized in LLMs. After that, we analyze the shortcomings of this objective in multi-step reasoning tasks like mathematical problems and design an additional training objective to alleviate this problem.

4.1 Unidirectional Modeling of Reasoning

The causal attention mechanism is widely applied to LLMs to fine-tune various downstream tasks, including multi-step mathematical reasoning problems. Formally, we denote a mathematical dataset as $D = (x_i, y_i)_{i=1}^{N}$ where x_i is a question, y_i represents the CoT reasoning steps to solve question x_i, N stands for the number of samples in D. For each sample, LLMs are trained to maximize the following likelihood:

$$\mathcal{L}_{\text{causal}} = \sum_{t}^{T} \log P(y_i^t | y_i^{<t}, x_i; \theta) \qquad (1)$$

where $y_i^{<t}$ denotes the previous tokens before token y_i^t, T denotes the length of y_i, and θ denotes the model parameters.

4.2 Bidirectional Modeling of Reasoning

Through applying Equation 1, LLMs strictly follow the order of deductions from left to right in the mathematical problems. Nevertheless, this training objective solely focuses on acquiring the association from conditions to conclusions, disregarding the reciprocal relationship. For example, as shown in Figure 2, LLMs are trained to predict the ultimate answer even the intermediate answer with preceding conditions, namely inferring step 4 from

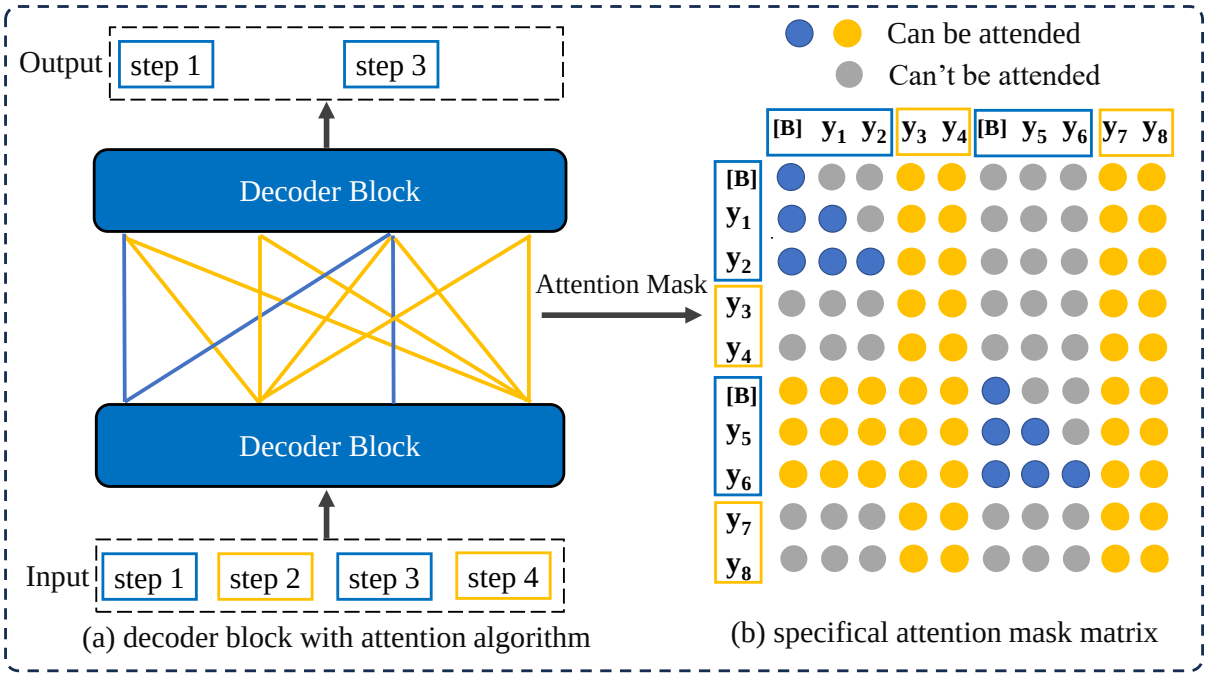

Figure 3: The left part (a) presents the decoder block with a modified attention algorithm. Steps 1 and 3 need to be predicted, which keep every token following a left-to-right order. Steps 2 and 4 are the observation steps that adopt bidirectional modeling. Sub-figure (b) is a specifical attention mask matrix for (a). [B] is a special token "BOS".

steps 1 to 3 and inferring step 3 from step 2, while disregarding the process of deducing preceding conditions from the conclusions, such as how to infer the deductions of numbers "13" and "14" appeared in the conclusion based on step 4. This shortcoming is essentially the lack of context modeling for reasoning steps, which causes insufficient dependency between different steps and affects the general reasoning performance. Struggling with the reversal questions is one of the typical manifestations.

To sufficiently model the relationship between different reasoning steps and improve their overall performance, especially the reversal reasoning ability, we propose a simple and effective training framework, which introduces an additional bidirectional training objective based on the characteristics of multi-step deduction. In particular, for each CoT explanation y_i, combined with n steps $y_i = \{s_1, s_2, ..., s_n\}$, we randomly sample parts of the steps that need to be predicted, denoted as s_i^{pred}, and the rest steps denoted as s_i^{obs}. Given that LLMs leverage the causal attention mechanism throughout the pre-training even fine-tuning phases, it proves challenging to directly transition the attention mechanism from unidirectional to bidirectional. To maintain the original capabilities of LLMs and better stimulate the ability of reversal reasoning, we keep each token in s_i^{pred} following a left-to-right order, which is shown in Figure 3 (b). Besides, s_i^{obs} could be observed by each token in s_i^{pred}, which achieves the bidirectional modeling of the preceding and following steps. At the same time, s_i^{pred} are not visible in s_i^{obs} in order to prevent information leakage. For instance, in Figure 3, if step 2 can obtain the information of step 1, it would lead to a potential information leakage that could impact the prediction of step 1. Formally, the proposed new training objective could be described as follows:

$$\mathcal{L}_{\text{bid}} = \sum_{t \in s_i^{pred}}^{T^{pred}} \log P(y_i^t | y_i^{<t} \cup y_i^{t \in s_i^{obs}}, x_i; \theta) \quad (2)$$

where T^{pred} denotes the number of tokens in s_i^{pred}.

4.3 Training and Inference

The causal and proposed bidirectional training objectives have been described in the previous section. Now, we clarify the final objective and the details of training and inference. In the training stage, we combine L_{causal} and L_{bid} as the final objective. Not only can it maintain the original capacity of LLMs, but it also improves the reversal reasoning ability. The corresponding computation can be formulated as follows:

$$\mathcal{L} = \mathcal{L}_{\text{causal}} + \alpha \mathcal{L}_{\text{bid}} \quad (3)$$

where α is a customized parameter. As shown in Figure 3 (b), to satisfy the autoregressive generation for each step in s_i^{pred}, the special token [BOS] is padded at the beginning of the input. At the stage of inference, LLMs still adopt the causal attention algorithm as usual to perform reasoning autoregressively.

5 Experiments

In this section, we evaluate the effectiveness of our method by employing it on mathematical datasets. To compare with current works (Fu et al., 2023a; Han et al., 2023; Yu et al., 2024), we follow them and adopt the original test set to present the general reasoning ability. The discussion about the reversal reasoning capacity will be introduced in Section 6.1.

5.1 Datasets

To evaluate LLMs' reasoning ability and general ability, we utilize four mathematical datasets, namely GSM8K (Cobbe et al., 2021), MultiArith (Roy and Roth, 2015), ASDiv (Miao et al., 2020), and SVAMP (Patel et al., 2021). Except for LLama2 (MetaMath) with and without our method, which uses the MetaMath training dataset, we only fine-tune Flan-T5 and LLama2 on the GSM8K training set, which contains 7,473 examples. It's noticed that for each GSM8K training instance, we employ GPT-3.5-Turbo-1106 to generate multiple related reasoning paths and choose the correct one as the final solution (specifical prompt could be seen in Appendix A). The remaining three datasets are adopted to evaluate the out-of-distribution ability of models. Moreover, following the previous work (Fu et al., 2023a; Han et al., 2023), we adopt 500 examples for each dataset as the validation set, and the remaining examples as the test set (800 for GSM8K, 400 for MultiArith, 18K for ASDiv, 500 for SVAMP).

5.2 Baselines

For the baseline models, we divide them into three categories: (i) Closed-sourced models: GPT-3.5-Turbo-1106 (OpenAI, 2022), Code-Davinci-002 (Chen et al., 2021a), LaMDA-137B (Kojima et al., 2022), PaLM-60B (Chowdhery et al., 2022), each of them presents strong reasoning ability. (ii) Open-sourced generic models: Flan-T5 (Chung et al., 2022) and LLama2 (Touvron et al., 2023), which are widely applied to various tasks. (iii) Specialized models: For Flan-T5 models, SpecialFT (Fu et al., 2023a) and DialCoT (Han et al., 2023) respectively employ knowledge transfer and questions decomposition to enhance models' mathematical reasoning ability. For LLama2 models, Rejection sampling Fine-Tuning (RFT) (Yuan et al., 2023) collects multiple correct reasoning paths as augmented data for fine-tuning. WizardMath (Luo et al., 2023a) applies reinforcement learning from the evol-instruct feedback method to the math domain. MetaMath (Yu et al., 2024) proposes four data augmentation methods, significantly improving performance. Besides, we apply the supervised fine-tuning (SFT) method on our designed GSM8K training dataset for both models.

5.3 Implementation

We implement our method on two model architectures, namely Encoder-Decoder (Flan-T5) and Decoder-only (LLama2). Due to the limited computing resources, we chose Flan-T5-3B and LLama2-7B as backbones and fully fine-tuned them. The greedy search algorithm is utilized to execute inference processes for all the specialized models. More experimental details can be seen in Appendix A.1. We follow the previous works to use statistical significance tests (Koehn, 2004) to detect if the difference in accuracy score between our approach and base settings is significant.

5.4 Results

The overall results are shown in Table 2 and can be summarized as follows:

Results on Flan-T5 backbone. The SFT 3B model trained on GPT-3.5-Turbo-1106 generated dataset is better than SpecialFT and DialCoT-S-PPO, even better than their 11B on parts of datasets, presenting the importance of data quality. Moreover, SFT and our method outperform LaMDA-137B on four datasets, and are superior to PaLM-60B, LLama2 7B to 13B on parts of datasets, showing the reasoning potential of the smaller models. Compared with the SFT method, which employs the causal attention mechanism, our method effectively improves the general reasoning ability of models, which achieves an improvement of 7.3% on GSM8K in testing accuracy. Besides, on three out-of-distribution datasets, applying our method also obtains significant improvements.

Results on LLama2 7B backbone. Our approach outperforms the RFT method, which collects more reasoning paths as augmented data for fine-tuning,

Methods	Backbone	#Params.	Math Word Problems			
			GSM8K	MultiArith	ASDiv	SVAMP
Closed-sourced models						
GPT-3.5-Turbo-1106	-	-	77.4	97.2	90.4	79.2
Code-Davinci-002		175B	63.1	95.8	80.4	76.4
Kojima et al. (2022)	LaMDA	137B	14.8	45.0	46.6	37.5
Chowdhery et al. (2022)	PaLM	60B	29.9	75.0	61.9	46.7
Open-sourced models						
Chung et al. (2022)	Flan-T5	3B	13.5	24.0	20.7	17.7
Chung et al. (2022)	Flan-T5	11B	16.1	51.7	36.5	39.7
Touvron et al. (2023)	LLama2	7B	13.7	45.2	50.1	33.4
Touvron et al. (2023)	LLama2	13B	25.3	64.8	59.0	43.2
Touvron et al. (2023)	LLama2	70B	52.1	92.5	74.7	67.4
Specialized models with Flan-T5						
SpecialFT (Fu et al., 2023a)	Flan-T5	3B	22.4	42.3	28.4	23.8
SpecialFT (Fu et al., 2023a)	Flan-T5	11B	27.1	63.0	37.6	35.6
DialCoT-S-PPO (Han et al., 2023)	Flan-T5	3B	25.6	46.9	30.7	27.1
DialCoT-S-PPO (Han et al., 2023)	Flan-T5	11B	37.1	68.1	40.9	41.7
SFT	Flan-T5	3B	28.0	59.2	48.8	38.8
SFT w/ Ours†	Flan-T5	3B	**35.3**	**69.3**	**54.3**	**53.6**
Specialized models with LLama2						
RFT (Yuan et al., 2023)	LLama2	7B	45.3	90.5	50.9	39.8
WizardMath (Luo et al., 2023a)	LLama2	7B	56.7	89.0	61.1	61.4
SFT	LLama2	7B	46.0	90.0	51.0	51.0
SFT w/ Ours†	LLama2	7B	**52.1**	**90.0**	**60.3**	**59.2**
MetaMath (Yu et al., 2024)	LLama2	7B	65.0	96.7	75.0	72.4
MetaMath w/ Ours†	LLama2	7B	**68.0**	**97.0**	**79.3**	**77.6**

Table 2: The accuracy of various LLMs on four mathematical datasets. † denotes that the performance improvements over standard SFT and MetaMath are statistically significant with $p < 0.05$.

and is close to WizardMath, which applies reinforcement learning and additional augmented data. Besides, after adopting our method on the MetaMath dataset, the general reasoning ability of the model is further enhanced, which obtains an average improvement of 3.2% in testing accuracy.

Overall results summary. The specialized mathematical datasets are important for models to enhance reasoning ability because all specialized models get significant improvements compared with their backbones. Moreover, applying our approach to different settings, such as the different backbones, they all achieve significant improvements. Finally, the above experiment results demonstrate that our method is beneficial to modeling the relationship of different CoT reasoning steps and improving the general mathematical reasoning ability.

6 Analysis

6.1 Evaluation of the Reversal Ability

To further evaluate the effectiveness of our method in improving the reversal reasoning ability, we conduct experiments on two reversal mathematical datasets designed previously. We directly adopt the models described in Section 5 to perform inference. As shown in Table 3, all specialized models tuned on the mathematical dataset, get improvements compared with their backbones. Compared with standard SFT, our method achieves obvious improvements, even outperforming RFT and close to WizardMath on LLama2-7B. It's noticed that

Models	**GSM8K-Rev**	**MathQA-Rev**
GPT-3.5-Turbo	52.2	44.6
Flan-T5-3B	3.5	5.8
Flan-T5-11B	12.3	9.6
Flan-T5-3B (SFT)	8.2	10.0
w /Ours	**17.4**	**13.1**
LLama2-7B	7.0	10.3
LLama2-7B (RFT)	20.8	14.5
LLama2-7B (WizardMath)	25.9	24.1
LLama2-7B (SFT)	21.7	16.5
w /Ours	**25.7**	**23.5**
LLama2-7B (MetaMath)	50.0	37.0
w /Ours	**55.7**	**40.3**

Table 3: The accuracy of various LLMs on reversal GSM8K and MathQA test datasets.

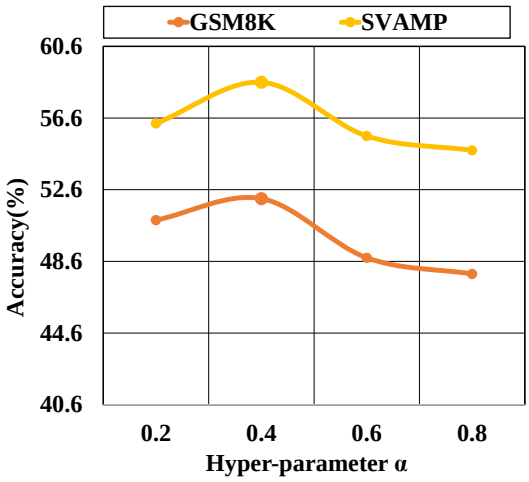

Figure 4: The effect of hyper-parameter α for LLama2-7B on GSM8K and SVAMP datasets.

MetaMath, using backward data augmentation, significantly improves the reversal reasoning ability, which presents the importance of reversal augmented data. After training with our method on the MetaMath dataset, LLama2-7B obtains higher accuracy and outperforms the strong baseline GPT-3.5-Turbo on GSM8K-Reversal. The above results further evaluate that our method is able to model the relationship between CoT reasoning steps better and further improve the reversal reasoning ability.

6.2 Effect of Hyper-parameter α

As described in Section 4.3, we introduce an additional training objective and combine it with the causal objective. To explore the effect of L_{bid} weight, we set the pre-defined hyper-parameter α from 0.2 to 0.8, train LLama2-7B on the GSM8K dataset, and evaluate it on the GSM8K and SVAMP datasets. As shown in Figure 4, our method achieves the best performance when α is 0.4. If α is too large, L_{bid} could damage the original causal paradigm and reduce the performance. On the contrary, once α becomes too small, the L_{bid} has no effect on training and can't provide effective context modeling.

6.3 Extensibility of Method

To evaluate the extensibility of our method, we conduct experiments on two commonsense reasoning datasets: CQA (Talmor et al., 2019) and QASC (Khot et al., 2020), which are respectively five-choice and eight-choice question-answering datasets. We denote each instance in datasets as (x, y, r), where x is a commonsense question

Models	Method	QASC	CQA
GPT-3.5-Turbo	-	62.1	77.2
T5-XL-3B	$x2y$	71.2	79.0
	$(x2y) + (x2r)$	73.0	79.6
	$(x2y) + (x2(y + r))$	74.8	81.1
	Ours	**76.0**	**82.6**

Table 4: The accuracy of T5-XL with different training methods on QASC and CQA datasets.

with multiple choices, y is its corresponding label, namely one of the correct choices, and r is a rationale that describes the knowledge of correct label with multi-step. We also leverage GPT-3.5-Turbo-1106 to generate a related rationale r for every sample. The specific prompt is shown in Appendix A.3.

Following Hsieh et al. (2023) and Li et al. (2022), we set the T5 model as the backbone, and denote the training method $[f(x) \rightarrow y]$ as $x2y$, $[f(x) \rightarrow y] + [f(x) \rightarrow y + r]$ as $(x2y) + (x2(y + r))$ (Li et al., 2022), $[f(x) \rightarrow y] + [f(x) \rightarrow r]$ as $(x2y) + (x2r)$ (Hsieh et al., 2023), where f is the training model. The above training methods are utilized as strong baselines. To sufficiently model the relationships between y and every step in r, we adopt the $(x2(y + r) + (x2(y + r)')$ method, where $(y + r)'$ denotes that employ our bidirectional modeling on $(y + r)$. Table 4 illustrates that r is beneficial to provide more knowledge to improve the performance. Besides, our method outperforms all the baselines, which demonstrates the effectiveness and generalization of our method.

7 Conclusion

In this paper, we discussed the "reversal curse" in mathematical problems and proposed two reversal datasets based on the GSM8K and MathQA to evaluate whether LLMs face challenges in reversal mathematical reasoning. We analyzed the potential reason and attributed it to the insufficient modeling of the relationship between reasoning steps. Consequently, based on the characteristics of multi-step deduction tasks, a bidirectional training objective is designed to alleviate this problem. Finally, we conducted experiments on four mathematical reasoning benchmarks to evaluate the effectiveness of our method and also verify the benefit of our approach on two reversal datasets we constructed. In the future, we will explore other enhanced methods and explore how to apply them in the pre-training stage to improve the general reasoning ability.

8 Limitation

In this section, we present several of the limitations of this paper. Firstly, the process of reversal data construction needs to be further refined, such as utilizing GPT-3.5 or GPT-4 to generate higher-quality data and verify their accuracy. Besides, our method designs an additional training objective which could increase the cost of computational resources. Finally, we have not yet applied our method to larger-scale models, such as LLama2-13B and LLama2-70B, due to the limitation of computational resources. We will further explore the performance of our method on these larger-scale models.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2357–2367. Association for Computational Linguistics.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2023. Learning from mistakes makes llm better reasoner.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: Llms trained on "a is b" fail to learn "b is a".

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and et al. 2020. Language models are few-shot learners. In Advances in neural information processing systems, pages 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, and et al. 2021a. Evaluating large language models trained on code.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, HenriquePondedeOliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, M.N. Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, FelipePetroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, WilliamH. Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, I. Babuschkin, Suchir Balaji, Shantanu Jain, WilliamS. Saunders, Christopher Hesse, AndrewN. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, MatthewM. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, McCandlish Sam, Ilya Sutskever, and Wojciech Zaremba. 2021b. Evaluating large language models trained on code. Preprint arXiv:2107.03374.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, HyungWon 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, and Liam Fe. 2022. Palm: Scaling language modeling with pathways.

HyungWon Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, ShixiangShane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, EdH. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, QuocV. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

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.

K. Collins, A. Jiang, S. Frieder, L. Wong, M. Zilka, U. Bhatt, T. Lukasiewicz, Y. Wu, J. Tenenbaum, W. Hart, T. Gowers, W. Li, A. Weller, and M. Jamnik. 2023. Evaluating language models for mathematics through interactions.

Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active prompting with chain-of-thought for large language models.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023a. Specializing smaller language models towards multi-step reasoning. In Proceedings of the 40th International Conference on Machine Learning, pages 10421–10430.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023b. Complexity-based prompting for multi-step reasoning. In Advances in International Conference on Learning Representations.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023c. Complexity-based prompting for multistep reasoning. In Advances in International Conference on Learning Representations.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamilė Lukošiūtė, Karina Nguyen, Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. 2023. Studying large language model generalization with influence functions.

Chengcheng Han, Xiaowei Du, Che Zhang, Yixin Lian, Xiang Li, Ming Gao, and Baoyuan Wang. 2023. Dial-CoT meets PPO: Decomposing and exploring reasoning paths in smaller language models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 8055–8068. Association for Computational Linguistics.

D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In Advances in Neural Information Processing Systems: Datasets and Benchmarks.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, volume 1: Long Papers, pages 14852–14882.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego De, Las Casas, Lisa Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, GeorgeVanDen Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pages 523–533.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In Findings of the Association for Computational Linguistics: ACL 2023, pages 8003–8017. Association for Computational Linguistics.

S. Imani, L. Du, and H. Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models.

Yan Junbing, Chengyu Wang, Taolin Zhang, Xiaofeng He, Jun Huang, and Wei Zhang. 2023. From complex to simple: Unraveling the cognitive tree for reasoning with small language models. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 12413–12425. Association for Computational Linguistics.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In In The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, pages 8082–8090.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Proceedings of the 2004 conference on empirical methods in natural language processing, pages 388–395.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In Advances in neural information processing systems, volume 35, pages 22199–22213.

Chengpeng Li, Zheng Yuan, Guanting Dong, Keming Lu, Jiancan Wu, Chuanqi Tan, Xiang Wang, and Chang Zhou. 2023a. Query and response augmentation cannot help out-of-domain math reasoning generalization.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, Wenhu Chen, and Xifeng Yan. 2022. Explanations from large language models make small reasoners better.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023b. Making language models better reasoners with step-aware verifier. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5315–5333. Association for Computational Linguistics.

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.

Weize Liu, Guocong Li, Kai Zhang, Bang Du, Qiyuan Chen, Xuming Hu, Hongxia Xu, Jintai Chen, and Jian Wu. 2023. Mind's mirror: Distilling self-evaluation capability and comprehensive thinking from large language models.

H. Luo, Q. Sun, C. Xu, P. Zhao, J. Lou, C. Tao, X. Geng, Q. Lin, S. Chen, and D. Zhang. 2023a. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct.

Z. Luo, C. Xu, P. Zhao, Q. Sun, X. Geng, W. Hu, C. Tao, J. Ma, Q. Lin, and D. Jiang. 2023b. Wizardcoder: Empowering code large language models with evol-instruct. Preprint arXiv:2306.08568.

Ang Lv, Kaiyi Zhang, Shufang Xie, Quan Tu, Yuhan Chen, Ji-Rong Wen, and Rui Yan. 2023. Are we falling in a middle-intelligence trap? an analysis and mitigation of the reversal curse.

Jun-Yu Ma, Jia-Chen Gu, Zhen-Hua Ling, Quan Liu, and Cong Liu. 2023. Untying the reversal curse via bidirectional language model editing.

LucieCharlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2022. Teaching small language models to reason. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, volume 2: Short Papers, pages 1773–1781.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing English math word problem solvers. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 975–984. Association for Computational Linguistics.

Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In Advances in International Conference on Learning Representations.

OpenAI. 2022. Gpt-3.5-turbo.

OpenAI. 2023. Gpt-4.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, K. Slama S. Agarwal, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Christiano P. Welinder, J. Leike, and R. Lowe. 2022. Training language models to follow instructions with human feedback. In Advances in neural information processing systems.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1743–1752. Association for Computational Linguistics.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. Distilling reasoning capabilities into smaller language models. In Findings of the Association for Computational Linguistics: ACL 2023, pages 7059–7073. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4149–4158. Association for Computational Linguistics.

R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. Koura, M. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. Smith, R. Subramanian, X. Tan, B. Tang, R. Taylor, A. Williams, J. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In Advances in the eleventh International Conference on Learning Representations.

Yue Wang, Hung Le, Akhilesh Gotmare, Nghi Bui, Junnan Li, and Steven Hoi. 2023. CodeT5+: Open code large language models for code understanding and generation. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 1069–1088. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In Advances in Neural Information Processing Systems, page 35:24824–24837.

Da Wu, Jingye Yang, and Kai Wang. 2023. Not all large language models (llms) succumb to the "reversal curse": A comparative study of deductive logical reasoning in bert and gpt models.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In Advances in the twelfth International Conference on Learning Representations.

Z. Yuan, H. Yuan, C. Li, G. Dong, C. Tan, and C. Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In Advances in International Conference on Learning Representations.

Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. 2023a. Progressive-hint prompting improves reasoning in large language models.

Qinkai Zheng, Xiao Xia, Xu Zou, Yuxiao Dong, Shan Wang, Yufei Xue, Lei Shen, Zihan Wang, Andi Wang, Yang Li, Teng Su, Zhilin Yang, and Jie Tang. 2023b. Codegeex: A pre-trained model for code generation with multilingual benchmarking on humaneval-x. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '23, page 5673–5684. Association for Computing Machinery.

A Appendix

A.1 Experiment Details

We adopt the AdamW (Kingma and Ba, 2014) optimizer to train all the models. Following Chung et al. (2022) and Yuan et al. (2023), we respectively fine-tune Flan-T5 and LLama2 models for 50 and 3 epochs, and set batch size as 64 and 128, learning rate 5e-5 and 1e-5. For LLama2 with different enhanced methods, we directly adopt the checkpoints provided in huggingface to perform inference. We run all the experiments on eight NVIDIA NVIDIA A100-PCIE-40GB.

A.2 Prompts for COT Reasoning Paths

The specific prompts for GPT-3.5-Turbo-1106 to obtain COT reasoning paths are shown in Figure 5.

A.3 Prompts for QA Reasoning

The specific prompts for GPT-3.5-Turbo-1106 to obtain QA rationales are shown in Figure 6.

A.4 Case study

To understand the effectiveness of our approach better, we provide an example of LLama2-7B with SFT and our approach on the GSM8K dataset. As shown in Figure 7, SFT doesn't sufficiently understand the relationships of different conditions in the question, which causes the wrong reasoning process. On the contrary, benefiting from the bidirectional modeling, our method understands the context and infers the correct answer.

Prompt:
You are a helpful and precise assistant, please follow the examples and answer the question with reasonin g procession, finally give that 'The answer is ' with calculating expressions.

Question: Paige was helping her mom plant flowers and together they planted some seeds. They put 10 seeds in each flower bed. If there are 45 flowerbeds\nHow many seeds did they plant?
Answer: They put 10 seeds in each flower bed. There are 45 flower beds. So they planted 10 * 45 seeds. 10 * 45 is 450. The answer is 450.

Question: Jack received 3 emails in the afternoon, 6 emails in the morning and some more in the evening . If he received a total of 10 emails in the day\nHow many emails did jack receive in the evening?
Answer: Jack received 3 emails in the afternoon, 6 emails in the morning and some more in the evening. If he received a total of 10 emails in the day, then he received 10 - (3 + 6) = 1 emails in the evening. The answer is (10 - (3 + 6)).

Question: At the arcade Dave had won 14 tickets and lost 2 tickets. If he used 10 to buy some toys\nHow many tickets did Dave have left?
Answer: Dave started with 14 tickets. He lost 2 tickets. So he had 14 - 2 = 12 tickets. He used 10 tickets to buy toys. So he had 12 - 10 = 2 tickets left. The answer is ((14 - 2) - 10).

Question: {Q}

Figure 5: The prompt for obtaining COT reasoning steps with a conclusion that “The answer is”.

Prompt:
You are a helpful and precise assistant, following the examples and just explaining the reason given the question and answer.

Question: How do you reduce pollution\nAnswer choices:\n(a) igniting fuel and oxidiser\n(b) transporta tion technology\n(c) wasting\n(d) not recycling\n(e) burning fossil fuels\n(f) converting electricity to he at\n(g) water conservation\n(h) using less resources\nThe answer is using less resources. Why?
Answer: Conserving resources has a positive impact on the environment. Use of resources affects the en vironment such as pollution.

Question: What will move to another area if their habitat will no longer support them? \nAnswer choices:\n(a) density\n(b) Birds\n(c) squids\n(d) humans\n(e) clouds\n(f) gravity\n(g) cows\n(h) Whales\nThe answer is cows. Why?
Answer: If a habitat can no longer support animals then those animals will move to another area. Cows are social animals.

Question: With the exception of allergies, what may cause a person to seek medical attention? \nAnswer choices:\n(a) Contact with latex\n(b) a tree falling\n(c) Organs within the body.\n(d) Contact with baby chicks\n(e) prolactin release\n(f) Contact with peanut butter\n(g) hypothyroidism\n(h) Contac t with microorganisms\nThe answer is Contact with microorganisms. Why?
Answer: Microorganisms can cause infections. Infections usually require medical treatment.

Question: {Q}\nThe answer is {Answer}. Why?
Answer:

Figure 6: The prompt for obtaining the commonsense reasoning steps based on the correct choice.

Question:
Below is an instruction that describes a task. Write a response that appropriately completes the request. Instruction: Carlos, Jim and Carrey were at the beach playing and they decided to gather some seashells. Jim collected 27 seashells, which was 5 more than what Carlos collected. Carlos collected twice as many as Carrey. They gathered all their seashells and divided them equally between themselves. How many did each person get? (The correct answer is 24)

SFT Answer:
First, we know that Carlos collected twice as many seashells as Carrey. So, if Carlos collected 27 seashell s, then Carrey collected 27 / 2 = 13.5 seashells. Next, we know that Jim collected 5 more seashells than C arlos, so he collected 27 + 5 = 32 seashells. Now, we have the total number of seashells collected by all o f them: 27 (Carlos) + 13.5 (Carrey) + 32 (Jim) = 72 seashells. Finally, we divide the total number of seas hells equally between themselves: 72 / 3 = 24 seashells for each person. The answer is 24.

Ours Answer:
We know that Jim collected 27 seashells, which was 5 more than what Carlos collected. So, Carlos collec ted 27 - 5 = 22 seashells. Carlos collected twice as many as Carrey, so Carrey collected 22 / 2 = 11 seashe lls. When they gathered all their seashells, they had a total of 27 + 22 + 11 = 60 seashells. If they divided them equally between themselves, each person would get 60 / 3 = 20 seashells. So, each person got 20 se ashells. The answer is (60 / 3).

Figure 7: A case study about the reasoning procession on the GSM8K dataset.