Turning Dust into Gold: Distilling Complex Reasoning Capabilities from LLMs by Leveraging Negative Data

Yiwei Li[1]*, Peiwen Yuan[1]*, Shaoxiong Feng[2], Boyuan Pan[2], Bin Sun[1], Xinglin Wang[1], Heda Wang[2], Kan Li[1]†

[1] School of Computer Science, Beijing Institute of Technology
[2] Xiaohongshu Inc
{liyiwei,peiwenyuan,binsun,wangxinglin,likan}@bit.edu.cn
{shaoxiongfeng2023,whd.thu}@gmail.com {panby}@zju.edu.cn

Abstract

Large Language Models (LLMs) have performed well on various reasoning tasks, but their inaccessibility and numerous parameters hinder wide application in practice. One promising way is distilling the reasoning ability from LLMs to small models by the generated chain-of-thought reasoning paths. In some cases, however, LLMs may produce incorrect reasoning chains, especially when facing complex mathematical problems. Previous studies only transfer knowledge from positive samples and drop the synthesized data with wrong answers. In this work, we illustrate the merit of negative data and propose a model specialization framework to distill LLMs with negative samples besides positive ones. The framework consists of three progressive steps, covering from training to inference stages, to absorb knowledge from negative data. We conduct extensive experiments across arithmetic reasoning tasks to demonstrate the role of negative data in distillation from LLM.

MATH Dataset	Intersection	Pos	Neg	IoU
InterAlgebra	4	35	21	0.077
Prealgebra	9	72	43	0.085
Geometry	1	21	10	0.033
NumberTheory	1	29	17	0.022
Precalculus	2	25	16	0.051
Probability	4	19	16	0.129
Algebra	8	52	43	0.062
Overall	29	253	166	0.074

Table 1: The distribution of correct answers in MATH test set. Pos and Neg refer to models trained on positive and negative samples respectively. Intersection over Union (IoU) exhibits a remarkably low value across all subsets, which confirms the value of negative samples.

Introduction

Nowadays, owing to chain-of-thought (CoT) prompting (Wei et al. 2022b), large language models (LLMs) exhibit strong reasoning capabilities (Bubeck et al. 2023a), especially when it comes to complex mathematical problems (Hendrycks et al. 2021). Unfortunately, CoT has been demonstrated to be an emergent property of models with more than 100B parameters, but not of smaller models (Wei et al. 2022a). The burdensome computational requirements and high inference costs of these models hinder their development in real-world scenarios with limited resources (Ho, Schmid, and Yun 2023). Thus, the goal of our research is to enable complex arithmetic reasoning in small models for deploying at scale.

Knowledge distillation (Hinton, Vinyals, and Dean 2015) offers a promising way to transfer specific capabilities from LLMs into smaller models. This process is also referred to as model specialization enforcing compact models to focus on certain skills. Prior works (Magister et al. 2023; Fu et al. 2023; Hsieh et al. 2023) employed LLMs with in-context learning (ICL) (Brown et al. 2020) to generate reasoning paths (rationales) of math problems, which are more beneficial for small models to acquire complex reasoning ability than reference reasoning paths. Table 1 shows an intriguing phenomenon: models trained on positive and negative data separately have an extremely small overlap (intersection) in their correct answers on the MATH test set. Although the negative model has a lower accuracy, it can address some questions that the positive model is unable to provide correct answers, which confirms the valuable knowledge contained in negative data. Additionally, the undesirable behaviors within negative data are also useful when preventing the model from committing similar issues. Another reason that we should exploit negative data is the token-based pricing strategy of OpenAI. Even for GPT-4, the accuracy on MATH dataset is less than 50% (Bubeck et al. 2023b), meaning that all tokens of negative data are charged for nothing. Therefore, instead of discarding negative samples, we extract and utilize valuable knowledge from negative samples to boost the model specialization.

The conventional process of model specialization can be summarized as three steps (Zhu et al. 2023): The first step is chain-of-thought distillation, training small models with reasoning chains generated from LLMs. The second step can be regarded as self-enhancement, conducting self-distillation (Mobahi, Farajtabar, and Bartlett 2020) or self-augmentation to further optimize the models. Besides, self-consistency (Wang et al. 2023) is widely used as an effective decoding

*Equal contributions.
†Corresponding author.

strategy to boost the model performance in reasoning tasks. In this work, we propose a novel model specialization framework[1] (shown in Figure 1) that can exploit negative data to enhance the distillation of the complex reasoning abilities from LLMs. Specifically, we first develop the negative assistant training (NAT) approach, where dual LoRA (Hu et al. 2022) structure is designed to capture knowledge from both positive and negative sides. As an auxiliary module, the knowledge of negative LoRA can be dynamically integrated into the training process of positive LoRA through a corrected attention mechanism. For self-enhancement, we devise negative calibrated enhancement (NCE), which regards the negative output as a baseline to strengthen the distillation of critical positive rationales. In addition to the training stage, we also leverage the negative information during inference. Traditional self-consistency allocates equal or probability-based weights to all candidate outputs, leading to some fallible answers being voted up. To alleviate this issue, adaptive self-consistency (ASC) is proposed to conduct ranking before voting, where the ranking model is trained on both positive and negative data.

We perform comprehensive experiments and detailed analyses across arithmetic reasoning tasks with LLaMA-7b (Touvron et al. 2023) as the student model. Previous model specialization work only validated on ordinary datasets (e.g., GSM8K, ASDiv, etc.), while we are the first to focus on the challenging competition mathematical problems – MATH dataset (Hendrycks et al. 2021). Experiments show that: (1) Negative assistant training can provide a more comprehensive way to absorb the knowledge from negative data. (2) Negative calibrated enhancement can make the process of self-distillation more targeted on crucial knowledge. (3) Ranking model trained on both positive and negative rationales can assign appropriate weights for answer aggregation. In summary, key contributions of this work are as follows:

- We illustrate that negative samples with incorrect answers can also provide a valuable resource besides positive data for distilling knowledge from LLMs in complex arithmetic reasoning tasks.
- To fully leverage the negative data, we propose a model specialization framework consisting of three progressive steps, spanning from training to inference stages.
- Extensive evaluations on challenging arithmetic reasoning dataset demonstrate that the proposed framework can effectively exploit the negative information and outperform baselines by a large margin.

Related Work

Chain-of-Thought Reasoning

The approach of solving complex reasoning problems by generating chain-of-thought (rationales) has been proven to be an effective method (Wei et al. 2022b). By following the pattern of gradually solving sub-problems, both few-shot CoT (Fu et al. 2023) and zero-shot CoT (Kojima et al. 2022) can stimulate the potential reasoning ability of LLMs. On this basis, Least-to-most prompting (Zhou et al. 2023) suggests explicitly splitting the problem and solving them step by step. Self-Consistency (Wang et al. 2023) further improves accuracy by conducting vote between multiple diverse rationales. PHP proposes (Zheng et al. 2023) iteratively generating answers and adding the historically generated answers as hints to the context to achieve the final convergence on the answer. Both correct and incorrect answers generated during this iteration process serve as hints to provide effective information. We also think that responses with incorrect answers from LLMs can provide valuable information, but the differences lie in: (1) We believe that not only the generated answers, but also the rationales contain valuable knowledge. (2) We consider utilizing these negative samples in the process of transferring knowledge from LLMs to smaller models instead of only inference stage.

Distilling Knowledge from Large Models

Knowledge distillation (Hinton, Vinyals, and Dean 2015; Sanh et al. 2019) has proven effective for transferring knowledge from a large model to a smaller one. This process is usually achieved by optimizing the parameters of smaller models so that their outputs (distributions (Feng et al. 2021), hidden states (Jiao et al. 2020), attentions (Tang et al. 2019)) can be closer to that of large models. However, the black-box nature of current mainstream LLMs (e.g., GPT4) hinders the application of these methods. Thus, many studies (Ho, Schmid, and Yun 2023; Fu et al. 2023; Zhu et al. 2023) have attempted to conduct hard distillation by fine-tuning smaller models directly on the LLMs generated responses with correct answers. However, as previously mentioned, responses generated by LLMs that contain incorrect answers also contain valuable knowledge. Discarding this portion of data directly would be a pity, especially considering that a significant portion of responses generated by LLMs in complex reasoning tasks end with incorrect answers. To this end, we propose multiple methods to fully utilize these *abandoned knowledge* in the process of transferring reasoning abilities of LLMs.

Learning from Negative Views

Samples that reflect some particular undesirable behavior are called negative data, which has been studied to help model correct such behavior (He and Glass 2020; Welleck et al. 2020; Lagutin, Gavrilov, and Kalaidin 2021). He and Glass (2020) conducts negative updates with training signals provided by negative samples to avoid model generating such data. Welleck et al. (2020); Li et al. (2020) penalizes the model for outputting words with certain characteristics by introducing an unlikelihood loss term. Li et al. (2022, 2023) suggests maximizing the distance between the predictions of the negative teacher and student. These methods only consider the use of negative training signals in negative samples. But in fact, negative data can also provide valuable positive knowledge. In this work, we investigated how to comprehensively utilize knowledge of negative data from both positive and negative perspectives.

[1]Our code and data have been released on https://github.com/Yiwei98/TDG.

Methodology

Background

Chain-of-Thought Distillation Rajani et al. (2019) demonstrated that training a language model on a dataset with explicit rationales preceding the answer could improve the ability to generate the final answer. Thus, chain-of-thought distillation is proposed to maximize the manifestation of the reasoning abilities of the LLMs on smaller models. Denote $\mathcal{D} = \{(x_i, y_i)\}^N$ to be a dataset with N training instances, where x_i is a problem and y_i is its answer. Given an additional set of M demonstrations $\{d_i = (x_i^d, r_i^d, y_i^d)\}^M$, where r represents rationales, the prompt $\{p_i = (d_1, ..., d_M, x_i)\}^N$ is input to the LLM and obtain responses $\{(\hat{r}_i, \hat{y}_i)\}^N$. Previous work (Ho, Schmid, and Yun 2023; Fu et al. 2023; Zelikman et al. 2022) retain positive samples S_{pos} where $\hat{y} = y$, and maximizes the likelihood of the student model to generate $(\hat{r}, \hat{y})$ as follows:

$$\mathbb{E}_{(x,\hat{r},\hat{y})\sim S_{pos}} \log P(\hat{y}, \hat{r}|x; \theta). \quad (1)$$

Self-Enhancement Based on the idea of human self reflection to achieve progress, various methods (Huang et al. 2022; Xu et al. 2021; Mobahi, Farajtabar, and Bartlett 2020) have been proposed to strengthen language models based on their own knowledge, which we collectively refer to as self-enhancement. It consists two common methods: one is self-augmentation (Xu et al. 2021), where the model first generates data with diversity and then trains on them to achieve better generalization (Huang et al. 2022). The other is self-distillation (Zhu et al. 2023), which involves using the model itself as teacher to complete iterative distillation, thereby utilizing dark knowledge to further improve the performance.

Self-Consistency Self-consistency (Wang et al. 2023) capitalizes on the notion that a intricate problem requiring logical thinking usually offers several distinct approaches that all lead to the same accurate answer. Based on this, multiple candidates $\{(\hat{r}^l, \hat{y}^l)\}^L$ to problem x are suggested to generate through sampling, and the most consistent $\hat{y}$ is selected as the final prediction through a voting process:

$$\hat{y} = \arg\max_i \sum_{l=1}^{L} \mathbb{I}(\hat{y}^l = i) \quad (2)$$

where $\mathbb{I}(\hat{y}^l = i)$ is the indicator function (equal to 1 when $\hat{y}^l$ is equal to answer i, and 0 otherwise).

Negative Assistant Training

As shown in Table 1, negative samples also contains valuable knowledge, which can even serve as a good complement to positive data. However, there is an increased risk of inference errors for $\hat{r}$ corresponding to negative data where $\hat{y} \neq y$. Extracting useful knowledge from negative samples without being affected by undesirable behaviors is therefore a challenging task. To address this, we propose a two-stage Negative Assistant Training (NAT) Paradigm (*Step 1.1 and 1.2* in Figure 1).

Absorbing Negative Knowledge First, we acquire the $(x, \hat{r}, \hat{y})$ triplets from the LLM on mathematical problems as described in Background. To facilitate comprehensive learning of diverse problem-solving approaches from the LLM, we collect 8 distinct responses for each question, and categorizing these samples into $\mathcal{D}_{pos}$ and $\mathcal{D}_{neg}$ based on whether $\hat{y}$ equals to $\hat{y}$. Directly finetuning LLaMA on the union of $\mathcal{D}_{pos}$ and $\mathcal{D}_{neg}$ will inevitably introduce undesirable behaviors into the model. Therefore, we consider training a negative model on $\mathcal{D}_{neg}$ first, and extracting useful knowledge from it afterwards. We choose LoRA module (Hu et al. 2022) for its parameter efficient characteristics to finetune LLaMA on $\mathcal{D}_{neg}$ by maximizing the following expectation:

$$\mathbb{E}_{(x,\hat{r},\hat{y})\sim \mathcal{D}_{neg}} \log P(\hat{y}, \hat{r}|x; \theta_{neg}). \quad (3)$$

During this process, the parameters of LLaMA remain frozen, while the knowledge of $\mathcal{D}_{neg}$ is absorbed by LoRA θ_{neg}. We denote LLaMA with θ_{neg} as $\mathcal{M}_{neg}$.

Dynamic Integration Unit Since it is impossible to predetermine which mathematical problems θ_{neg} excels at, we design Dynamic Integrate Unit as shown in Figure 2 to dynamically integrate knowledge from θ_{neg} during the learning process of positive knowledge in $\mathcal{D}_{pos}$. We freeze θ_{neg} to prevent the knowledge inside from being forgotten and additionally introduce positive LoRA modules θ_{pos}. In each layer of LLaMA, we denote the output values obtained from θ_{pos} (θ_{neg}) as h_{pos} (h_{neg}) for input hidden states $h_{input} \in \mathbb{R}^d$. Ideally, we should positively integrate h_{pos} and h_{neg} to complement the beneficial knowledge in $\mathcal{D}_{neg}$ with respect to $\mathcal{D}_{pos}$ if h_{neg} contains beneficial knowledge. When h_{neg} contains detrimental knowledge, we should negatively integrate h_{pos} and h_{neg} to assist in reducing the possible undesirable behaviors within $\mathcal{D}_{pos}$. We propose a corrected attention mechanism to achieve this vision as follows:

$$\alpha = W_Q(h_{input}) W_K([h_{pos}; h_{neg}])^T + [0.5; -0.5] \quad (4)$$

$$h_{output} = \alpha \cdot W_V([h_{pos}; h_{neg}])) \quad (5)$$

where $W_Q, W_K \in \mathbb{R}^{d\times w}$ and $W_V \in \mathbb{R}^{d\times d}$ are trainable parameters during this stage (we use a bottleneck structure for W_V for reducing parameters). In Eq. (4), we use h_{input} as the query to calculate attention weights for h_{pos} and h_{neg}. By adding a correction term [0.5;-0.5] on $[\alpha_{pos}; \alpha_{neg}]$, the attention weights for h_{neg} are constrained to the range of [-0.5, 0.5], thus achieving the effect of adaptively integrating knowledge from h_{neg} in both positive and negative directions. Ultimately, the sum of h_{output} and the LLaMA layer outputs forms the output of the Dynamic Integrate Unit.

By employing NAT, $\mathcal{M}_{NAT}$ can inherit LLM's knowledge more comprehensively in both dimensions of diversity (more samples) and type (both positive and negative data), leading to improved complex reasoning abilities.

Negative Calibrated Enhancement

To further strengthen the reasoning ability of the model, we propose Negative Calibrated Enhancement (NCE) that

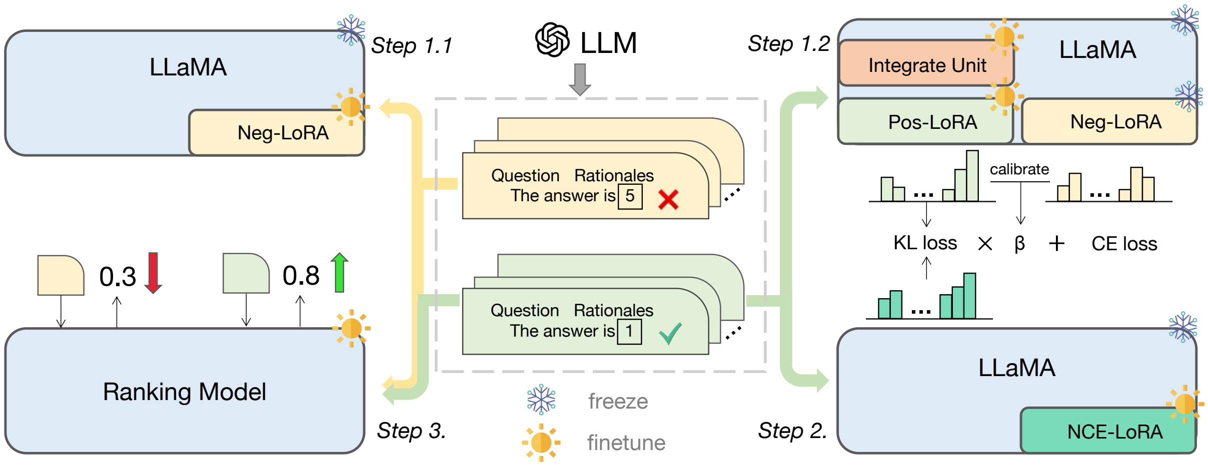

Figure 1: The overview of proposed framework. Step 1: Training Neg-LoRA on negative samples to assist in the learning of reasoning on positive data through Integrate Unit. Step 2: Utilizing Neg-LoRA as baseline to calibrate the process of self-enhancement. Step 3: Training a ranking model on both positive and negative samples. Then weighting the candidates adaptively during inference according to scores from it.

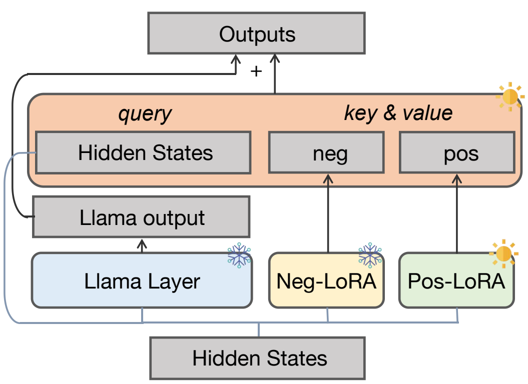

Figure 2: The workflow of Integrate Unit. The outputs of both Neg-LoRA and Pos-LoRA are fused through a corrected attention mechanism.

use negative knowledge to aid with the self-enhancement process (Zhu et al. 2023; Huang et al. 2022). We first use $\mathcal{M}_{NAT}$ to generate n $(\hat{r}, \hat{y})$ pairs for each problem in $\mathcal{D}$ as augmentation samples and supplement them to $\mathcal{D}_{pos}$. As for self-distillation (Zhu et al. 2023; Liu et al. 2020), we notice that some samples may contain more critical reasoning steps that can distinguish reasoners with different abilities. Our primary objective is to identify these key rationales and enhance the learning of them during self-distillation. Considering that $\mathcal{M}_{NAT}$ has incorporated the valuable knowledge from $\mathcal{M}_{neg}$, the key insights that make $\mathcal{M}_{NAT}$ a superior reasoner than $\mathcal{M}_{neg}$ are implicit in the inconsistent rationale generating distributions between the two. Therefore, we use KL divergence to measure such inconsistency and maximize the expectation in Eq. (8), where θ_{NCE} denotes the LoRA module of the student model $\mathcal{M}_{NCE}$.

$$f_{\text{KL}}(\theta_1, \theta_2) = \text{KL}(P(\hat{r}, \hat{y}|x; \theta_1), P(\hat{r}, \hat{y}|x; \theta_2)) \tag{6}$$

$$\beta = \text{Tanh}(f_{\text{KL}}(\theta_{neg}, \theta_{NAT})) \tag{7}$$

$$\mathbb{E}_{(x,\hat{r},\hat{y})\sim S_{pos}} \log P(\hat{y}, \hat{r}|x; \theta_{NCE}) + \beta * f_{\text{KL}}(\theta_{NAT}, \theta_{NCE}) \tag{8}$$

A larger β value indicates a greater divergence between $\mathcal{M}_{NAT}$ and $\mathcal{M}_{neg}$, implying that the generating distribution of $\mathcal{M}_{NAT}$ contains more crucial knowledge. By introducing β to adjust the loss weights of different samples, $\mathcal{M}_{NCE}$ will be able to learn selectively and generalize the knowledge embedded in $\mathcal{M}_{NAT}$.

Adaptive Self-Consistency

Self-Consistency (SC) technique (Wang et al. 2023) is effective for further improving the performance of models in complex reasoning (Zhu et al. 2023). However, current methods either naively assign equal weights to each candidate or simply assign weights based on generation probabilities. These strategies fail to adjust the weights of candidates based on the quality of $(\hat{r}, \hat{y})$ during the voting phase, which could potentially obscure the correct candidates. To this end, we propose Adaptive Self-Consistency (ASC), which utilizes $\mathcal{D}_{neg}$ and $\mathcal{D}_{pos}$ to train a Ranking Model $\mathcal{M}_{rank}$ that can adaptively reweight candidates with justification.

Ranking Model Training. Ideally, we hope that $\mathcal{M}_{rank}$ assigns higher weights to rationales that lead to the correct answer and vice versa. Thus, we construct training samples $\{(p_i, q_i)\}^N$ in the following way:

$$(p, q) = \begin{cases} ([x, \hat{y}, \hat{r}], 1) & if \;\; \hat{y} = y \\ ([x, \hat{y}, \hat{r}], 0) & if \;\; \hat{y} \neq y \end{cases} \tag{9}$$

and use MSE loss to train $\mathcal{M}_{rank}$:

$$\mathcal{L}_{RM} = \sum_{i=1}^{N} \|\mathcal{M}_{rank}(p_i) - q_i\|_2 \tag{10}$$

Models	Methods	Counting& Probability	Inter Algebra	Number Theory	Precalculus	Prealgebra	Geometry	Algebra	Average
PaLM 62B	Few-shot	-	-	-	-	-	-	-	4.4
PaLM 540B	Few-shot	-	-	-	-	-	-	-	8.8
GPT3 175B	Few-shot	4.7	4.4	4.4	4.0	7.7	3.1	6.0	5.2
GPT3 13B	Fine-tune	4.1	4.7	5.5	5.8	6.8	7.1	5.3	5.6
LLaMA 7B	Fine-tune	2.96	3.58	2.96	3.85	4.61	3.46	4.56	3.88 +0%
GPT-3.5 Turbo CoT	CoT KD	4.15	4.17	5.37	4.58	8.82	4.54	4.61	5.29 +36.3%
	MIX	3.49	1.43	1.67	1.46	5.27	2.59	4.05	3.03 -21.9%
	CL	4.64	3.93	5.74	4.03	7.39	2.51	5.98	5.16 +33.0%
	NT	3.93	3.93	6.30	2.20	6.69	4.10	5.17	4.48 +15.4%
	UL	4.98	3.86	5.37	3.85	6.70	4.10	5.27	4.96 +27.8%
	NAT	5.70	5.24	6.67	3.85	9.99	5.64	7.94	6.81 +75.5%
GPT-4 CoT	CoT KD	3.71	4.88	6.30	3.30	6.56	3.67	7.73	5.59 +44.1%
	MIX	3.28	2.86	2.96	4.21	5.45	3.55	6.66	4.49 +15.7%
	CL	4.15	3.67	5.00	3.11	7.90	5.43	5.98	5.24 +35.1%
	NT	3.28	2.46	4.07	3.85	8.92	6.05	5.97	5.14 +32.4%
	UL	4.15	3.46	6.67	3.11	8.67	5.18	8.25	6.03 +55.4%
	NAT	6.11	4.65	5.56	4.58	8.50	4.92	9.78	6.83 +76.0%

Table 2: Experimental results (%) on MATH test set for NAT. We report the accuracy (solving rate) of math problems for each test set. Average is the mean value of all subjects. GPT3 and PaLM are from Hendrycks et al. (2021) and Lewkowycz et al. (2022), respectively. Comparing with standard fine-tune, NAT achieves about 75.75% increase.

Weighting Policy. Building upon the foundation of $\mathcal{M}_{rank}$, we revise Eq. (2) to Eq. (11) to achieve the vision of adaptively reweighting the candidates reasonably.

$$\hat{y} = \arg\max_{i} \sum_{l=1}^{L} \mathbb{I}(\hat{y}^l = i) \cdot \mathcal{M}_{rank}([x, \hat{y}^l, \hat{r}^l]) \tag{11}$$

From the view of knowledge transfer, ASC achieves further utilization of knowledge (positive and negative) embedded in LLMs to help smaller models attain better performance.

Experiments

Experimental Setup

This work focuses on the challenging competition mathematical dataset MATH (Hendrycks et al. 2021), which has 12,500 problems in total spanning seven various subjects. Besides, the following four datasets are introduced to evaluate the generalization ability on out-of-distribution (OOD) data of the proposed framework: GSM8K (Cobbe et al. 2021), ASDiv (Miao, Liang, and Su 2020), MultiArith (Roy and Roth 2015), and SVAMP (Patel, Bhattamishra, and Goyal 2021).

For teacher model, we use gpt-3.5-turbo and gpt-4 API from OpeanAI to synthesize reasoning chains. Given that the problems of MATH are challenging, we select LLaMA-7b as the student model.

There are two main types of baselines in our study: one includes LLMs, while the other is based on LLaMA-7b. In the case of LLMs, we compare with two popular models: GPT3 (Brown et al. 2020) and PaLM (Chowdhery et al. 2022). As for LLaMA-7b, we first provide a comparison of our method with three settings: Few-shot, Fine-tune (on original training samples), CoT KD (chain-of-thought distillation). In terms of learning from negative views, four baseline methods will be further included: MIX (directly trains LLaMA with the mixture of both positive and negative data), CL (contrastive learning), NT (negative training) (He and Glass 2020) and UL (unlikelihood) (Welleck et al. 2020). The evaluations of NCE and ASC will also include some other baselines that will be introduced in corresponding parts.

Main Results

Native Assistant Training The evaluation results of NAT are presented in Table 2, with all methods using greedy search (i.e. temperature = 0). It shows that proposed method NAT improves task accuracy across all baselines. It can be seen from the low values of GPT3 and PaLM that MATH is a very difficult math dataset, but NAT can still accomplish competitive performance with much less parameters. Comparing with fine-tuning on the original data, NAT achieves about 75.75% increase under two different CoT sources. In comparison with CoT KD on positive samples, the mainstream specialization pattern, NAT also improves accuracy significantly, demonstrating the value of negative samples. As for baselines to utilize negative information, the lowest performance of MIX suggests that directly training the negative samples will make model toxic. Other methods are also mostly inferior to NAT, which indicates that using negative samples only in the negative direction is not sufficient in complex reasoning tasks.

Negative Calibrated Enhancement The main results under the data of gpt-3.5-turbo of NCE are shown in Figure 3. Compared with knowledge distillation (KD), NCE achieves an average progress of 10% (0.66), which demonstrates the effectiveness of distillation with calibration in-

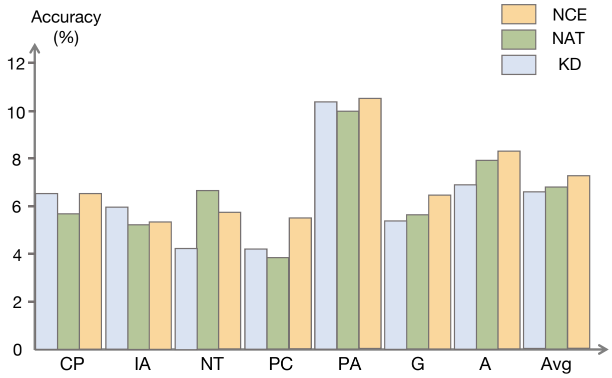

Figure 3: Experimental results (%) of NCE. KD denotes knowledge distillation with data augmentation.

Models	Strategies	CP	NT	PC	A*	Ave
CoT KD	SC	7.38	6.62	5.70	8.70	7.85
	SC *w*WS	7.22	6.64	5.75	8.52	7.82
	ASC	7.70	6.97	6.16	9.12	8.25
NAT	SC	8.65	7.49	5.75	11.14	9.25
	SC *w*WS	8.67	7.34	5.77	11.08	9.21
	ASC	9.30	8.33	5.83	11.88	9.84
NCE	SC	9.21	7.94	5.96	11.32	9.69
	SC *w*WS	9.13	7.84	5.99	11.25	9.64
	ASC	9.87	8.21	6.37	11.89	10.23

Table 3: Experimental results (%) on MATH for ASC. A* is the average of InterAlgebra, Prealgebra and Algebra.

formation offered by negative samples. Although NCE reduced some parameters (e.g., Neg-LoRA) compared to NAT, it still achieved a progress of 6.5% (0.44), implementing compressed model and improved performance.

Adaptive Self-Consistency To evaluate ASC, we compare it with base SC and its weighted sum (WS) version. We generate 16 samples with sampling temperature $T = 1$. The results from Table 3 shows that ASC is a more promising strategy to aggregate the answers from different samples. SC with WS doesn't outperform base SC, which is consistent with Wang et al. (2023). Note that the accuracy of ranking model is only about 60%, indicating that the performance of ASC can be further improved with higher accuracy. Refer to Accuracy of Ranking Model for detailed analysis.

Analysis

In order to better understand the usefulness of the negative knowledge and the effectiveness of our framework, we carry out extensive analysis on LLaMA distilled from gpt-3.5-turbo in terms of both quantitative and qualitative measures.

Generalization Besides MATH dataset, we evaluate the generalization ability of proposed framework. Following Fu et al. (2023), we only synthesize data and train the models on GSM8K and evaluate on all the four test sets. The higher performance of NAT and NCE on GSM8K indicates that the proposed method can generalize to previously commonly used dataset in the field of model specialization. NCE outperforms others in A-M-S datasets suggests that the calibrated dark knowledge from logits distributions can improve out-of-distribution (OOD) performance.

Methods	GSM8K	ASDiv	MultiArith	SVAMP
Fine-tune	17.51	36.37	53.17	17.90
CoT KD	38.81	76.43	83.5	47.40
NAT	41.24	76.11	84.67	47.20
KD	41.55	75.86	88.05	50.70
NCE	41.93	77.67	88.67	51.50

Table 4: Generalization evaluation results (%).

Methods	CP	NT	PC	G	A*	Ave
NAT	5.70	6.67	3.85	5.64	7.72	6.81
- Neg Data	6.55	4.63	4.40	3.97	7.88	6.58
- Neg LoRA	5.02	4.81	5.31	4.75	7.27	6.05
- Att	2.84	5.19	4.21	4.38	5.99	5.22
- Dual	7.38	5.37	4.21	4.80	6.87	6.30

Table 5: Ablation study results (%) for NAT.

Ablation study To demonstrate the necessity of each component in NAT, we take a series of ablation study by removing the following parts: (1) Neg Data: The whole dual LoRA structure and attention integration only on positive data. (2) Neg LoRA: Based on (1), the negative LoRA will be further removed. (3) Att: Instead of the attention mechanism, we integrate two LoRA modules by a gated function. (4) Dual: We modify the range of Equation 4 to [0, 1] rather than [-0.5, 0,5], which means the knowledge from negative LoRA can only be absorbed from positive way.

The results are shown in Table 5. When filtering negative samples with same model structure, we find that model accuracy decreases, confirming the value of negative knowledge. Further removing the negative LoRA illustrates the importance of dual LoRA structure. The performance drops dramatically without attention mechanism, indicating it plays an important role in integrating LoRA modules. When changing the range of α to [0, 1], which forces positive LoRA to add the knowledge from negative LoRA without the minus option. The lower accuracy suggests that avoiding being influenced by undesirable behaviors while extracting useful knowledge from negative samples is necessary.

Attention To fully comprehend how knowledge from $\mathcal{M}_{neg}$ is integrated into $\mathcal{M}_{NAT}$ during the NAT process, we analyzed the averaged attention weights on the output of θ_{neg} (α_{neg} in Eq. (4)) along 3 dimensions: token position, question level, and question subject.

As shown in the Figure 4, as the position of the generated token increases, α_{neg} gradually decreases from positive values to negative values. This indicates that in the initial generation stage of the rationale, $\mathcal{M}_{neg}$ provides positive knowledge that helps establish the overall direction of problem solving. While in the fine-grained step generation

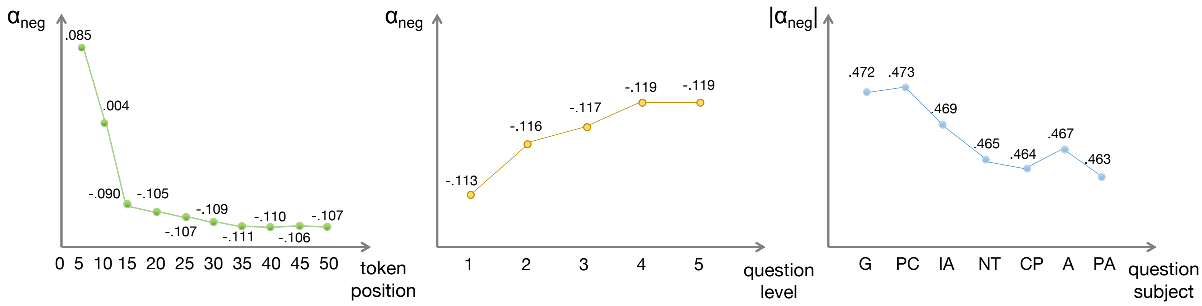

Figure 4: Analysis of α_{neg} along dimensions of: token position, question level, and question subject.

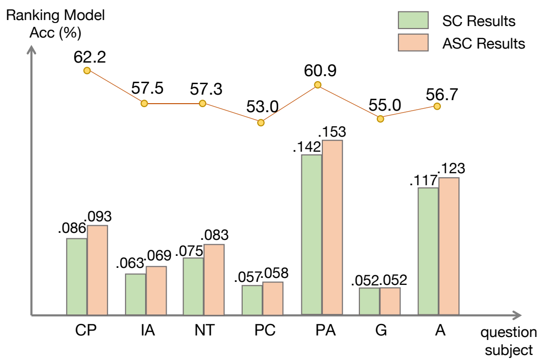

Figure 5: Relationship between the accuracy of distinguishing positive and negative rationales of $\mathcal{M}_{rank}$ and the improvement brought by ASC.

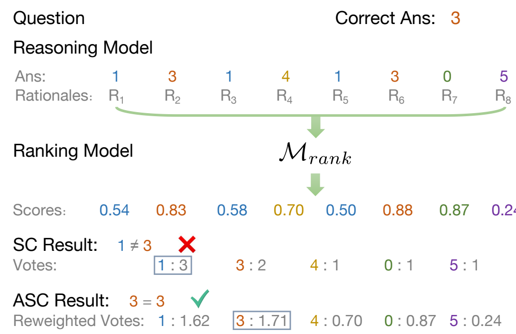

Figure 6: An intuitive example shows the strength of ASC.

stage, the undesirable behaviors provided by $\mathcal{M}_{neg}$ helps avoid similar errors. As for the question level, we observed that as the difficulty of the questions increases, the value of α_{neg} decreases. We hypothesize that this is because the undesirable behaviors in $\mathcal{M}_{neg}$ makes it challenging to accurately perform complex reasoning, leading it to primarily assist reasoning in difficult questions by providing negative references. Finally, we sort the subjects according to the accuracy (ascending order) of $\mathcal{M}_{NAT}$ as the X-axis coordinates of the right subgraph of Figure 4 to observe the tendency of $|\alpha_{neg}|$. Basically, $|\alpha_{neg}|$ gradually decreases with the difficulty of the subjects. Considering that the smaller $|\alpha_{neg}|$ is, the lesser the role $\mathcal{M}_{neg}$ plays in the NAT process ($|\alpha_{neg}| = 0$ indicating that $\mathcal{M}_{neg}$ is not involved in the NAT phase), we believe that $\mathcal{M}_{neg}$ can play a greater role in addressing challenging subjects during NAT.

Accuracy of Ranking Model We further explore the relationship between $\mathcal{M}_{rank}$'s accuracy of distinguishing positive and negative samples and the performance of ASC across 7 subjects. As shown in Figure 5, we observed a positive correlation among the accuracy of $\mathcal{M}_{rank}$, SC results, and the improvement of ASC over SC. One potential reason is that relatively easier subjects can lead the reasoning model to achieve higher reasoning accuracy. This not only contributes to improved SC results but also provides more positive samples for training $\mathcal{M}_{rank}$. Through more comprehensive training, $\mathcal{M}_{rank}$ achieves higher accuracy, subsequently enhancing ASC's performance compared to SC. As current accuracy of $\mathcal{M}_{rank}$ is only around 60% and yet it can significantly enhance the effectiveness of SC, we believe that there is substantial room for improvement in ASC.

Case Study about Adaptive Self-Consistency We provide an intuitive example (Figure 6) to show the superiority of ASC. In this example, deceptive candidate 1 is chosen as the prediction by SC due to having more votes than the correct candidate 3. $\mathcal{M}_{rank}$ adjusts the weights of the eight candidates based on rationales, resulting in the reweighted correct candidate 3 obtaining a higher vote count.

Conclusion

This work explores the effectiveness of negative data for distilling the complex reasoning ability from large language models to specialized small ones. We propose a novel framework, consisting of three progressive steps and fully leveraging the negative information through the entire process of model specialization. NAT can provide a more comprehensive way to employ the negative information from two aspects. NCE is able to calibrate the process of distillation, making it more targeted on crucial knowledge. Ranking model trained on two views of rationales can assign appropriate weights for answer aggregation to achieve adaptive self-consistency. Extensive experiments demonstrate that our framework can improve the effectiveness of distilling reasoning ability by the generated negative samples.

Acknowledgments

This work is supported by Beijing Natural Science Foundation (No.4222037, L181010).

References

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y. T.; Li, Y.; Lundberg, S. M.; Nori, H.; Palangi, H.; Ribeiro, M. T.; and Zhang, Y. 2023a. Sparks of Artificial General Intelligence: Early experiments with GPT-4. *CoRR*, abs/2303.12712.

Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y. T.; Li, Y.; Lundberg, S. M.; Nori, H.; Palangi, H.; Ribeiro, M. T.; and Zhang, Y. 2023b. Sparks of Artificial General Intelligence: Early experiments with GPT-4. *CoRR*, abs/2303.12712.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; Schuh, P.; Shi, K.; Tsvyashchenko, S.; Maynez, J.; Rao, A.; Barnes, P.; Tay, Y.; Shazeer, N.; Prabhakaran, V.; Reif, E.; Du, N.; Hutchinson, B.; Pope, R.; Bradbury, J.; Austin, J.; Isard, M.; Gur-Ari, G.; Yin, P.; Duke, T.; Levskaya, A.; Ghemawat, S.; Dev, S.; Michalewski, H.; Garcia, X.; Misra, V.; Robinson, K.; Fedus, L.; Zhou, D.; Ippolito, D.; Luan, D.; Lim, H.; Zoph, B.; Spiridonov, A.; Sepassi, R.; Dohan, D.; Agrawal, S.; Omernick, M.; Dai, A. M.; Pillai, T. S.; Pellat, M.; Lewkowycz, A.; Moreira, E.; Child, R.; Polozov, O.; Lee, K.; Zhou, Z.; Wang, X.; Saeta, B.; Diaz, M.; Firat, O.; Catasta, M.; Wei, J.; Meier-Hellstern, K.; Eck, D.; Dean, J.; Petrov, S.; and Fiedel, N. 2022. PaLM: Scaling Language Modeling with Pathways. *CoRR*, abs/2204.02311.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *CoRR*, abs/2110.14168.

Feng, S.; Ren, X.; Li, K.; and Sun, X. 2021. Multi-View Feature Representation for Dialogue Generation with Bidirectional Distillation. In *AAAI 2021*, 12812–12820.

Fu, Y.; Peng, H.; Ou, L.; Sabharwal, A.; and Khot, T. 2023. Specializing Smaller Language Models towards Multi-Step Reasoning. *CoRR*, abs/2301.12726.

He, T.; and Glass, J. R. 2020. Negative Training for Neural Dialogue Response Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 2044–2058. Association for Computational Linguistics.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Hinton, G. E.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *CoRR*, abs/1503.02531.

Ho, N.; Schmid, L.; and Yun, S. 2023. Large Language Models Are Reasoning Teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, 14852–14882. Association for Computational Linguistics.

Hsieh, C.; Li, C.; Yeh, C.; Nakhost, H.; Fujii, Y.; Ratner, A.; Krishna, R.; Lee, C.; and Pfister, T. 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, Toronto, Canada, July 9-14, 2023*, 8003–8017. Association for Computational Linguistics.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Huang, J.; Gu, S. S.; Hou, L.; Wu, Y.; Wang, X.; Yu, H.; and Han, J. 2022. Large Language Models Can Self-Improve. *CoRR*, abs/2210.11610.

Jiao, X.; Yin, Y.; Shang, L.; Jiang, X.; Chen, X.; Li, L.; Wang, F.; and Liu, Q. 2020. TinyBERT: Distilling BERT for Natural Language Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, 4163–4174. Association for Computational Linguistics.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large Language Models are Zero-Shot Reasoners. In *NeurIPS*.

Lagutin, E.; Gavrilov, D.; and Kalaidin, P. 2021. Implicit Unlikelihood Training: Improving Neural Text Generation with Reinforcement Learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, 1432–1441. Association for Computational Linguistics.

Lewkowycz, A.; Andreassen, A.; Dohan, D.; Dyer, E.; Michalewski, H.; Ramasesh, V. V.; Slone, A.; Anil, C.; Schlag, I.; Gutman-Solo, T.; Wu, Y.; Neyshabur, B.; Gur-Ari, G.; and Misra, V. 2022. Solving Quantitative Reasoning Problems with Language Models. In *NeurIPS*.

Li, M.; Roller, S.; Kulikov, I.; Welleck, S.; Boureau, Y.; Cho, K.; and Weston, J. 2020. Don't Say That! Making Inconsistent Dialogue Unlikely with Unlikelihood Training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10,* *2020*, 4715–4728. Association for Computational Linguistics.

Li, Y.; Feng, S.; Sun, B.; and Li, K. 2022. Diversifying Neural Dialogue Generation via Negative Distillation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, 407–418. Association for Computational Linguistics.

Li, Y.; Feng, S.; Sun, B.; and Li, K. 2023. Heterogeneous-Branch Collaborative Learning for Dialogue Generation. In *AAAI 2023*, 13148–13156.

Liu, W.; Zhou, P.; Wang, Z.; Zhao, Z.; Deng, H.; and Ju, Q. 2020. FastBERT: a Self-distilling BERT with Adaptive Inference Time. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 6035–6044. Association for Computational Linguistics.

Magister, L. C.; Mallinson, J.; Adámek, J.; Malmi, E.; and Severyn, A. 2023. Teaching Small Language Models to Reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, 1773–1781. Association for Computational Linguistics.

Miao, S.; Liang, C.; and Su, K. 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, ACL 2020, Online, July 5-10, 2020*, 975–984. Association for Computational Linguistics.

Mobahi, H.; Farajtabar, M.; and Bartlett, P. L. 2020. Self-Distillation Amplifies Regularization in Hilbert Space. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Patel, A.; Bhattamishra, S.; and Goyal, N. 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, NAACL-HLT 2021, Online, June 6-11, 2021*, 2080–2094. Association for Computational Linguistics.

Rajani, N. F.; McCann, B.; Xiong, C.; and Socher, R. 2019. Explain Yourself! Leveraging Language Models for Commonsense Reasoning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 4932–4942. Association for Computational Linguistics.

Roy, S.; and Roth, D. 2015. Solving General Arithmetic Word Problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 1743–1752. The Association for Computational Linguistics.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Tang, R.; Lu, Y.; Liu, L.; Mou, L.; Vechtomova, O.; and Lin, J. 2019. Distilling Task-Specific Knowledge from BERT into Simple Neural Networks. *CoRR*, abs/1903.12136.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR*, abs/2302.13971.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q. V.; Chi, E. H.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; Chi, E. H.; Hashimoto, T.; Vinyals, O.; Liang, P.; Dean, J.; and Fedus, W. 2022a. Emergent Abilities of Large Language Models. *Trans. Mach. Learn. Res.*, 2022.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E. H.; Le, Q. V.; and Zhou, D. 2022b. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *NeurIPS*.

Welleck, S.; Kulikov, I.; Roller, S.; Dinan, E.; Cho, K.; and Weston, J. 2020. Neural Text Generation With Unlikelihood Training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xu, X.; Wang, G.; Kim, Y.; and Lee, S. 2021. AugNLG: Few-shot Natural Language Generation using Self-trained Data Augmentation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, 1183–1195. Association for Computational Linguistics.

Zelikman, E.; Wu, Y.; Mu, J.; and Goodman, N. D. 2022. STaR: Bootstrapping Reasoning With Reasoning. In *NeurIPS*.

Zheng, C.; Liu, Z.; Xie, E.; Li, Z.; and Li, Y. 2023. Progressive-Hint Prompting Improves Reasoning in Large Language Models. *CoRR*, abs/2304.09797.

Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q. V.; and Chi, E. H. 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Zhu, X.; Qi, B.; Zhang, K.; Long, X.; and Zhou, B. 2023. PaD: Program-aided Distillation Specializes Large Models in Reasoning. *CoRR*, abs/2305.13888.