MinT: Boosting Generalization in Mathematical Reasoning via Multi-view Fine-tuning

Zhenwen Liang[1], **Dian Yu**[2], **Xiaoman Pan**[2], **Wenlin Yao**[2], **Qingkai Zeng**[1], **Xiangliang Zhang**[1], **Dong Yu**[2]
[1]University of Notre Dame [2]Tencent AI Lab, Bellevue
{zliang6,qzeng,xzhang33}@nd.edu
{yudian,xiaomanpan,wenlinyao,dyu}@global.tencent.com

Abstract

Reasoning in mathematical domains remains a significant challenge for relatively small language models (LMs). Many current methods focus on specializing LMs in mathematical reasoning and rely heavily on distilling knowledge from powerful yet inefficient large LMs (LLMs). In this work, we explore a new direction that avoids over-reliance on LLM teachers, introducing a multi-view fine-tuning method that efficiently exploits the generalization among mathematical problem datasets with diverse annotation styles. Our approach uniquely considers the various annotation formats as different "views" that may help each other and leverage them in training the model. By postpending distinct instructions to input questions, models can learn to generate solutions in diverse formats in a flexible manner. Experimental results show that our strategy enables relatively small LMs to outperform prior approaches that heavily rely on knowledge distillation, as well as carefully established baselines. Additionally, the proposed method grants the models promising generalization ability across various views and datasets, and the capability to learn from inaccurate or incomplete noisy data. We hope our multi-view training paradigm could inspire future studies in other machine reasoning domains.

Keywords: Large Language Models, Mathematical Reasoning

1. Introduction

Mathematical reasoning, a central aspect of human cognition, has been the subject of inquiry across various disciplines such as philosophy, mathematics, and cognitive science. This capacity, characterized by the analysis of symbolic patterns and logical relationships, and the derivation of conclusions from evidence, is crucial for numerous practical applications, such as intelligent education systems (Tack and Piech, 2022). The recent development of large language models (LLMs) (Brown et al., 2020; Ouyang et al., 2022; Touvron et al., 2023a) introduces both a novel challenge and an exciting opportunity for deep learning models to tackle mathematical reasoning tasks.

A significant advancement in machine mathematical reasoning is the discovery of a step-by-step reasoning process such as scratchpads (Nye et al., 2021) and chain-of-thought (CoT) prompting (Wei et al., 2022b; Kojima et al., 2022) to enhance reasoning in LLMs, leading to marked improvements in the accuracy of automated math problem solving. However, this strong ability seems to emerge only at an immense scale, typically exceeding 100 billion parameters (Wei et al., 2022a) in LLMs. Similar observations are presented in (Touvron et al., 2023a), revealing that LMs with fewer than 10 billion parameters still struggle to achieve accuracy over 20% on the GSM8K dataset (Cobbe et al., 2021), which is essentially comprised of elementary-level math word problems.

To obtain mathematical reasoning models that are both efficient and effective, a widely explored direction is to specialize general-purpose LMs in mathematics (Fu et al., 2023) by supervised fine-tuning and distilling the knowledge and abilities from larger teacher models into smaller student models (Ho et al., 2022; Shridhar et al., 2022; Magister et al., 2022; Hsieh et al., 2023; Liang et al., 2023b). However, this kind of approach faces certain limitations. Firstly, it heavily relies on CoT explanations of existing data or extra CoT-style data generated by the larger models to train the smaller student model, and the most common choices for teachers are the GPT series and PaLM-540B (Chowdhery et al., 2022), which are resource-intensive and costly. Moreover, LLMs might still make errors or fail to sufficiently explain reasoning steps, which could adversely influence the quality of the generated data and subsequently, the performance of the student models.

To mitigate the above limitations, instead of relying solely on inefficient LLMs to generate CoT annotations or additional training samples, we focus on an under-explored question:

Can we effectively utilize publicly accessible datasets to develop small LMs specialized in mathematical problem solving?

[*] This work was done when Zhenwen and Qingkai were interns at the Tencent AI Lab, Bellevue. The code and data are available at `https://github.com/Zhenwen-NLP/MinT-COLING2024`.

LREC-COLING 2024, pages 11307–11318
20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

Using existing annotated datasets can reduce manual effort and computational costs compared with relying on LLMs to generate additional annotated data. However, there are also several challenges posed by this direction. First, existing datasets vary significantly in their annotation formats. For instance, the GSM8K (Cobbe et al., 2021) dataset offers solutions in a narrative format detailing step-by-step rationales, while the MathQA (Amini et al., 2019) dataset uses flattened programs for annotations, and Ape210K (Zhao et al., 2020), the largest math word problem dataset, adopts equation-based solutions. Additionally, as we collect more data from various data sources such as websites, the potential of encountering irrelevant or even inaccurate data cannot be disregarded. These differences in annotation styles and quality can hinder the effective use of these datasets to train math reasoning models. Empirically, we observe that merely merging multiple datasets with different annotation formats cannot always improve model performance — in fact, it often has a negative effect.

To address the above challenges, we propose a **M**ulti-V**i**ew Fi**n**e-**T**uning (MinT) paradigm. In this context, the disparate annotation methods employed across different datasets are conceptualized as distinct "views" of mathematical problem solutions and thus we enable the generalization among different annotations during training to improve math understanding and reasoning. To fully leverage existing data and demonstrate generalization, we not only utilize the original views but also expand the solution views in existing math word problem datasets by view transformation. Then we append view-specific instructions to the input questions to guide the models to generate solutions in the desired view. Our underlying assumption is that training the model to comprehend various solution views equates to learning different methods of mathematical reasoning, which inherently helps strengthen its reasoning and generalization capabilities. Extensive experimental results support the efficacy of MinT, indicating that it fosters a variety of generalizations that contribute to enhancing overall performance across all views. Notably, our paradigm can also be used to incorporate relevant but noisy datasets, by regarding them as a new view, to further improve the performance of existing views.

In fact, some instruction-tuning methods (Chung et al., 2022; Sanh et al., 2022; Liu et al., 2023) for LLMs share similarities with our work in augmenting instructions and input-output pairs. However, our approach, MinT, primarily seeks to investigate the impact of various alternative reasoning paths for a single input. In contrast, these methods concentrate on generating and employing a broader range of high-quality instructions and question-answer pairs. To put it differently, while prior research emphasizes data generation, our proposed MinT focuses on the efficient utilization of existing data.

Our contributions can be summarized as follows:

- We propose a multi-view training approach to fine-tune a relatively small language model in the domain of mathematical reasoning. As a result, our approach boosts the performance on four mathematical reasoning benchmarks. Importantly, this is achieved without the use of LLMs as additional data generators or teacher models.
- Our multi-view training method utilizes a large amount of data from both the research community and the broader internet, to enhance the mathematical reasoning capabilities of LMs, which confirms great flexibility and generalizability by effectively handling and learning from data in diverse formats and from various sources.
- Our extensive experiments demonstrate that our approach performs effectively not only on an externally held-out dataset but also across different LM architectures. This insight demonstrates this approach can be applied more widely, and its potential to inspire future studies aimed at more diverse tasks and backbone models.

2. Related Work

2.1. Multi-View Learning

In traditional machine learning, multi-view learning often refers to semi-supervised co-training algorithms (Nigam and Ghani, 2000; Sun and Jin, 2011). These algorithms exploit multiple views of data to iteratively learn separate classifiers, each of which provides predicted labels for the unlabeled data of the others, i.e., semi-supervised setting. Another thread of multi-view learning lies in clustering methods. These methods aim to partition the data across multiple views, which provide complementary information to each other, to obtain a more refined representation of the data (Bickel and Scheffer, 2004; Li et al., 2015; Cao et al., 2015).
More recently, the concept of multi-view learning has been extended to deep learning (Song et al., 2020). For instance, Wang et al. (2022) employ multiple transformers to learn a comprehensive embedding for speaker recognition. Similarly, Zhong et al. (2023) utilize views based on context, syntax, and knowledge to analyze the sentiment of sentences. Moreover, Allen-Zhu and Li (2023) show that learning multi-view data (e.g., tail, legs, heads of a horse) can be associated with ensemble learning and knowledge distillation techniques to improve the accuracy of image classification tasks.
While traditional multi-view learning approaches aim to improve data representations, our work takes a different path by focusing on mathematical problem solving, in contrast to deterministic tasks such

as image classification, sentiment analysis, and speaker recognition. This free-form generation task can be characterized by its diverse solution views shown in Table 1. This situation is further complicated by the existence of multiple acceptable solutions for a given problem, as discussed in (Hong et al., 2021). Given these conditions, our goal is not to prioritize any single view. Instead, we focus on improving the accuracy of solutions across all views. In this context, the term "generalization" indicates the collaboration and mutual benefit across all the views, differing from the traditional concept of generalizing to a specific representation or improving the performance of deterministic tasks. We hope this work can broaden the application and impact of multi-view learning techniques in natural language processing.

2.2. Math Word Problem Solving

Solving math word problems is a representative task for evaluating the mathematical reasoning capabilities of NLP models (Amini et al., 2019; Patel et al., 2021; Cobbe et al., 2021). Earlier approaches rely on statistical and rule-based parsing (Hosseini et al., 2014; Koncel-Kedziorski et al., 2015), followed by a transition to Seq2Seq-based neural networks (Xie and Sun, 2019; Zhang et al., 2020; Jie et al., 2022). Recently, large language models have demonstrated success in solving math word problems, surpassing fine-tuned baselines through prompting methods such as CoT (Wei et al., 2022b; Kojima et al., 2022).

Another thread of research has also explored the distillation of data from LLMs to smaller models (Ho et al., 2022; Magister et al., 2022; Shridhar et al., 2022; Hsieh et al., 2023; Liang et al., 2023b; Yuan et al., 2023; Luo et al., 2023; Yue et al., 2023). These papers primarily leverage stronger LLMs for generating high-quality instructions and reasoning steps, segmenting problems, or creating customized exercises to train smaller models.

Our research, however, investigates a different approach: using accessible datasets in an effective and efficient way to train smaller LMs for mathematical problem solving, which can be seamlessly integrated with previous data-centric methods. In other words, we offer a unique perspective, being orthogonal to previous research efforts.

3. Our Approach

3.1. Our Views

Our method utilizes multi-view training where we conceptualize different annotation styles across datasets as distinct "views" of mathematical problem solutions. These views embody a rich collection of solution formats expressed in different levels of symbolism, each with unique nuances and strengths. We categorize the views as follows and show examples for the first three views in Table 1.

Clean Chain-of-Thought Explanations ($\mathbf{CoT_{clean}}$) The first view, **clean c**hain-**o**f-**t**hought explanations ($\mathrm{CoT_{clean}}$), is featured in the GSM8K dataset. This annotation style entails a thorough, step-by-step explanation of the solution process. Each intermediary step is clearly elaborated until the final solution is derived. These explanations serve as a detailed guide, illustrate the logical reasoning behind each step, and facilitate the comprehension of the entire solving process.

Equation Solutions (EQN) The second view, **eq**uatio**n** solutions (EQN), presents each question's solution as an equation composed of various operators and quantities. Although this view lacks the detailed explanation as those in CoT solutions, it offers a concise representation of the solution and is widely used in datasets such as Ape210K, MathQA, and CM17K. It provides the key information to solving problems in a form of a mathematical expression, making itself an efficient and effective format to address certain types of problems such as math word problems.

Solution Tree Pre-order Traversal (Tree) The third view, solution **tree** pre-order traversal (Tree), is an abstract representation of the solution. This format, which is widely adopted by math word problem solvers (Zhang et al., 2020; Liang et al., 2022a; Jie et al., 2022), employs the pre-order traversal of the solution tree. Using Tree eschews the need for parentheses, which in turn simplifies the solution grammar compared with EQN solutions. More importantly, this form reflects a goal-driven solving strategy aligned with human reasoning (Xie and Sun, 2019) as well as fosters efficient solution processing and inference.

Noisy Chain-of-Thought Explanations ($\mathbf{CoT_{noisy}}$) The fourth view, **noisy** chain-of-thought explanations ($\mathrm{CoT_{noisy}}$), is similar to $\mathrm{CoT_{clean}}$, albeit with noise introduced. This noise may come from incomplete explanations, minor calculation errors, irrelevant domains, or misinterpretation of the problem. This view represents a general category of irrelevant or inaccurate solutions, thereby cannot be used for evaluation and we do not provide examples in Table 1. While challenging, this view reflects the uncertainty and ambiguity in real-world data, providing an opportunity to make models more robust and flexible to different data sources.

In summary, the $\mathrm{CoT_{clean}}$view provides a detailed explanation, making it the richest in information. It outlines each solution step, mimicking the human method of detailed problem-solving. In contrast, the EQN view is more concise. It captures only

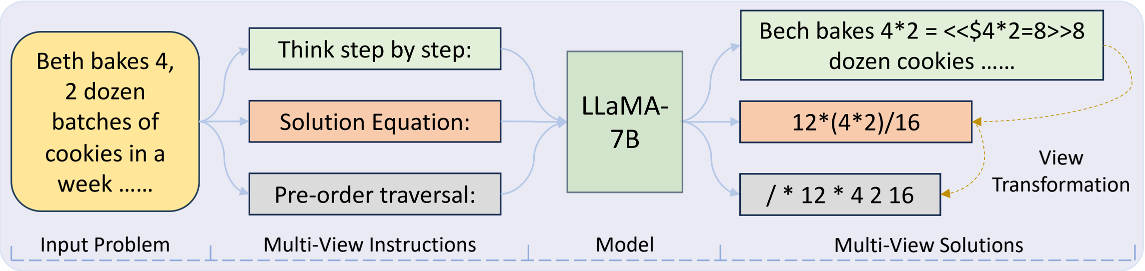


Figure 1: We use MinT to fine-tune a LLaMA-7B model that specializes in math problem solving. First, the original annotation is transformed into multiple different views. Then, the model is trained by instructions to generate different solution forms for one problem.

Question: Beth bakes 4, 2 dozen batches of cookies in a week. If these cookies are shared amongst 16 people equally, how many cookies does each person consume?

View	Solution
$\mathrm{CoT}_{\mathrm{clean}}$	Beth bakes 4, 2 dozen batches of cookies for a total of 4*2 = ≪4*2=8≫8 dozen cookies. There are 12 cookies in a dozen and she makes 8 dozen cookies for a total of 12*8 = ≪12*8=96≫96 cookies. She splits the 96 cookies equally amongst 16 people so they each eat 96/16 = ≪96/16=6≫ 6 cookies.
EQN	x = 12*(4*2)/16
TREE	/ * 12 * 4 2 16

Table 1: Examples of three views of mathematical solutions to a given question.

the core symbols necessary for the solution, ideal for quick interpretation and computation. Then, the TREE view further simplifies equation representation using a hierarchical approach, which is more concise and coherent. Our multi-view learning framework leverages these different solution forms. By training the model on multiple views, it gains a broader and deeper understanding. This is similar to how teachers encourage students to consider multiple solution paths to understand problems better. As a result, our approach can enhance the model's problem-solving capabilities.

3.2. View Transformation

As shown in Table 1, the $\mathrm{CoT}_{\mathrm{clean}}$ view contains both equations and explanations. Therefore, we can simply extract all the equations from the $\mathrm{CoT}_{\mathrm{clean}}$ view using rule-based detection, and then we combine and transform them into the EQN view. Apart from that, the third view, TREE, can be derived from the EQN view, through well-defined algorithms such as (Wang et al., 2018). This kind of view transformation allows us to augment the solution forms for the model to learn.

3.3. Multi-View Fine-Tuning

We propose an approach, referred to as **M**ulti-V**i**ew Fi**n**e-**T**uning (MinT 🌿), to guide the model in generating different views of solutions by postpending specific instructions to the input questions. This results in multiple unique concatenated instructions for a given problem, and each instruction guides the model to produce a corresponding view of the answer, as shown in Figure 1.

Formally, each question Q is paired with an instruction string p_i drawn from the set $\mathcal{P}$, which includes all possible instructions. When Q is concatenated with p_i, it results in a unique string for each question. This provides the necessary guidance for the model to generate a corresponding answer a_i from the answer set $\mathcal{A}$. As such, for each question, we formulate multiple sequences $s_i = Q + p_i + a_i$. Consequently, during the training phase, the model processes a large number of these sequences s_i, enhancing its understanding and generalization across multiple views.

To optimize the model, the next-word-prediction loss L is calculated for each sequence:

$$L(s_i) = -\sum_{j=1}^{len(s_i)-1} \log P(s_{i,j+1}|s_{i,1:j};\Theta), \tag{1}$$

where P denotes the model's conditional probability distribution over the next token, facilitated by the Softmax function of the model's logits, $s_{i,j}$ represents the j^{th} token of sequence s_i, and Θ embodies the model parameters. However, to enhance the model's focus on generating accurate answers, we exclusively backpropagate the loss calculated on the answer part, denoted as L_{a_i}:

$$L_{a_i}(s_i) = -\sum_{j=len(Q+p_i)+1}^{len(s_i)-1} \log P(s_{i,j+1}|s_{i,1:j};\Theta). \tag{2}$$

It ensures that the model focuses on learning to produce precise answers. During the evaluation, we adopt the same instruction and assess the model's performance on each individual view.

To sum up, our approach integrates multiple datasets, each following its own annotation format or "view". Our strategy of view transformation serves as an efficient data-utilization method to enhance reasoning generalization, which is achieved by converting data, originally annotated in one view, into multiple diverse views, thus maximizing the utilization of available data. Then, by learning from different views, our model can comprehensively understand mathematical problems, thus improving its reasoning and generalization capabilities. Another advantage of our method is its scalability. MinT can easily accommodate additional views or data, thus continually enhancing the model's learning and performance as new data becomes available. Our MinT also has the potential to be integrated with existing knowledge distillation-based methods (Ho et al., 2022; Shridhar et al., 2022; Magister et al., 2022; Hsieh et al., 2023; Liang et al., 2023b) and instruction tuning methods (Luo et al., 2023; Yue et al., 2023). In this scenario, the output from larger "teacher" models can be considered as an additional view. Our model, serving as the "student", can then learn to imitate the reasoning processes and outcomes of the teacher model, thus effectively expanding its own problem solving capability. This ability can be confirmed by the performance improvement given by the inclusion of ASDiv-CoT dataset in Section 4.3.

4. Experiments and Results

Our experiments mainly aim at answering the following research questions:

***RQ1**: How does MinT affect the performance of a mathematical reasoning model trained across different datasets with different views impact the model's performance in comparison to straightforward dataset merging?*

***RQ2**: What is the effect of introducing additional noisy training data on the model's generalization capabilities?*

***RQ3**: How does MinT affect the performance of a mathematical reasoning model trained on a single dataset when in comparison to individual fine-tuning on a single view?*

In addition to three main RQs, we evaluate performance on held-out datasets and experiment with different LM backbones to verify generalizability and adaptability in Section 4.4 and 4.7.

4.1. Datasets and Implementation

GSM8K ***($\mathrm{CoT_{clean}}$, EQN, TREE)*** The GSM8K dataset (Cobbe et al., 2021) is a curated set of 8.5K high-quality elementary-level math word problems in English, authored by human problem writers. It is split into approximately 7.5K problems for training and 1K for testing purposes. The problems are annotated with their comprehensive step-by-step solutions, providing the Clean Chain-of-Thought Explanations ($\mathrm{CoT_{clean}}$) view.

MathQA ***(EQN, TREE)*** MathQA (Amini et al., 2019) contains English mathematical problems from GRE examinations. Nevertheless, some of the problems in this dataset have quality concerns. Several efforts (Tan et al., 2022; Li et al., 2022; Liang et al., 2022a) have been conducted to cleanse and filter the MathQA dataset. We adopt the version referenced in (Liang et al., 2022a), wherein all solutions are re-annotated by an equation composed of the four arithmetic operators and numbers, reflecting the Equation Solutions (EQN) view and we also transform that to the TREE view.

Ape210K ***(EQN, TREE)*** The Ape210K dataset (Zhao et al., 2020) is a large-scale, template-rich collection of math word problems (MWPs) in Chinese, containing 210,488 problems and 56,532 solution templates. The view of the solutions in Ape210K mirrors that in MathQA. Our experiment incorporates its 200K training problems and 50K testing problems.

CM17K ***(EQN)*** The CM17K dataset (Qin et al., 2021) comprises four types of Chinese MWPs (arithmetic, one-unknown linear, one-unknown nonlinear, equation set), which is different from MathQA and Ape210K. Therefore, we only have the EQN view for the solutions in this dataset.

ASDiv-CoT ***($\mathrm{CoT_{noisy}}$)*** The ASDiv dataset (Miao et al., 2020) consists of 2,305 English MWPs that are diverse in language patterns and problem types. We employ the few-shot CoT predictions of GPT-3 provided by (Wei et al., 2022b) on this dataset as one of the $\mathrm{CoT_{noisy}}$views for training. With an accuracy of 71.3%, approximately 30% of the predictions are spurious. The inclusion of this dataset shows the adaptability and broad applicability of our method to inaccurate LLM-generated data.

ExamQA ***($\mathrm{CoT_{noisy}}$)*** The ExamQA dataset (Yu et al., 2021) is a comprehensive Chinese dataset of real-world exams, containing 638k multiple-choice instances across various subjects (e.g., sociology, education, and psychology). We sample a subset with 20k problems that contain numbers and equations in their answers by hand-crafted rules. Despite each problem in this subset being annotated with its ground truth and step-by-step solutions, we inevitably introduce many problems that are less relevant to the math subject. This dataset also serves as one of the $\mathrm{CoT_{noisy}}$ views, showing the generalizability of our approach.

We employ LLaMA-7B (Touvron et al., 2023a) as our backbone and perform fine-tuning on the full model. We use Pytorch with DeepSpeed Library to implement the code and use 8 NVIDIA V100 GPUs with 32GB of memory to train our model. We have incorporated a few techniques to ease the

	Evaluation Set							
	GSM8K (en)			MathQA (en)		CM17K (zh)	Ape210K (zh)	
Single Dataset Baselines								
Prior Best	38.2[a]			76.6[b]		54.1[c]	70.2[d]	
Trained on Single Dataset	35.4			79.9		70.1	74.0	
Simple Dataset Mixture								
Trained on GSM8K+MathQA	36.7			79.7		-	-	
Trained on Ape210K+CM17K	-			-		76.0	74.9	
Trained on All Four Datasets	35.3			81.0		68.9	74.1	
Multi-V**i**ew Fi**n**e-**T**uning (MinT ☘)								
	CoT$_{\text{clean}}$†	EQN	Tree	EQN†	Tree	EQN†	EQN†	Tree
Trained on GSM8K+MathQA	36.8	35.8	38.1	79.7	80.5	-	-	-
Trained on Ape210K+CM17K	-	-	-	-	-	77.1	75.9	74.3
Trained on All Four Datasets	38.8	39.2	**40.8**	81.0	**81.3**	**77.6**	**76.0**	74.3

Table 2: Experimental results showing the performance of LLaMA-7B with different fine-tuning methods across four datasets. The simple dataset mixture means the training data is mixed only with their original views (marked with †). For CM17K, we only adopt the EQN view due to the complexity of view transformation for this dataset. For our method MinT, we train our model and report the performance on all available views. The first column shows the training datasets that are used. Prior best results are: a: (Magister et al., 2022), b: (Liang et al., 2022a), c: (Qin et al., 2021), d: (Zhao et al., 2020). Some prior best models are based on RNNs, hence not as good as single dataset fine-tuning on LLaMA-7B. Missing cells in this table are due to language differences.

	GSM8K			MathQA		CM17K	Ape210K	
Simple Dataset Mixture								
Trained on Four Datasets	35.3			81.0		68.9	74.1	
Trained on Four Datasets + Two Noisy Datasets	31.9			79.7		71.3	73.2	
Multi-V**i**ew Fi**n**e-**T**uning (MinT ☘)								
	CoT$_{\text{clean}}$†	EQN	Tree	EQN†	Tree	EQN†	EQN†	Tree
Trained on Four Datasets	38.8	39.2	40.8	81.0	81.3	77.6	76.0	74.3
Trained on Four Datasets + ASDiv-CoT	39.0	39.7	42.2	81.4	81.8	78.2	76.4	75.2
Trained on Four Datasets + ExamQA	38.6	38.8	41.0	81.0	82.2	78.1	76.6	75.4
Trained on Four Datasets + Two Noisy Datasets	39.2	39.7	**42.4**	82.0	**82.3**	**78.8**	**77.0**	76.1

Table 3: Experimental results showing the performance of LLaMA-7B with different fine-tuning methods. Four datasets indicate the combination of four clean datasets - GSM8K, MathQA, CM17K, and Ape210K, while two noisy datasets are ASDiv-CoT and ExamQA.

computational burden. First, we apply parameter offloading and optimizer offloading and utilize gradient checkpointing to reduce the memory footprint. Additionally, we employ gradient accumulation, effectively enlarging the batch size without demanding additional GPU memory. Lastly, the parameters' precision is set to float-16 (FP16). The fine-tuning process lasts for three epochs with a batch size of 64 and a learning rate of 0.00002. Though we find that further fine-tuning could bring slight performance improvements, we limit the number of epochs to four due to efficiency concerns. As a result, the total training time, even while using all the 6 training sets, does not exceed 48 hours.

4.2. Generalization Across Different Datasets with Different Views (RQ1)

For this investigation, we utilize four different datasets: GSM8K, MathQA, CM17K, and Ape210K. Our baseline for comparison involves prior best results, the single dataset fine-tuning on LLaMA-7B, and simply merging all four datasets for fine-tuning on LLaMA-7B, as shown in Table 2. The results show that straightforward merging cannot bring any improvements. Contrastively, it even has a negative effect. Alternatively, with our multi-view learning approach applied to the four datasets, the model obtains a general improvement across all views when additional training data is added. Another notable observation is that the performance of the CoT$_{\text{clean}}$ view on the GSM8K dataset gets improved by multi-view fine-tuning with the other three datasets, even though the additional datasets actually do not provide any supplementary data in the CoT$_{\text{clean}}$ view. This outcome shows a promising generalization ability, illustrating the effectiveness of MinT in better leveraging diverse datasets. We can also notice that the performance on Tree view is generally the best, where the potential reason is that this view has the simplest grammar, hence it is the easiest view for the model to learn.

Furthermore, we observed that the accuracy improvement between the single dataset baseline and our method is more obvious on the GSM8K and CM17K in comparison to the MathQA and Ape210K. A possible explanation could be that MathQA and Ape210K already contain a substantial number of training problems, thereby enabling the learning of problem patterns and solving skills directly from their training sets. Consequently, the contribution of external datasets may not be significant in this case. However, for the more challenging GSM8K and CM17K datasets, our multi-view training could improve the performance more effectively. Furthermore, it can be observed that the EQN view performs optimally on the Ape210K dataset, which is different from GSM8K. This could potentially be attributed to the fact that the solutions in Ape210K comprise fewer steps, resulting in relatively simpler equations compared to those in GSM8K and MathQA. Consequently, converting these equations into tree traversals may not substantially simplify the solutions, thereby not improving the model performance. The above two behavior patterns are also shown in Table 3.

4.3. Generalization on $\mathrm{CoT}_{\mathrm{noisy}}$ View (RQ2)

In order to further understand the effects of incorporating external noisy training data, we introduce two additional datasets – ASDiv-CoT and ExamQA. The former provides CoT explanations to problems within the ASDiv dataset, though about 30% of these CoTs are incorrect. The latter, ExamQA, provides CoT to multi-subject exam problems, and while the solutions provided are accurate, a large number of them are less related to mathematical reasoning. Table 3 presents our experimental results: when we directly add the two noisy datasets for training, there is a slight decrease in accuracy. However, with a specific postfix to differentiate them from the other three views, the overall performance is improved, which demonstrates the potential of using external noisy data to improve the performance on specific downstream tasks. In addition, the results in Table 2 and 3 indicate that multilingual data can also complement each other and help improve the general reasoning ability.

4.4. Generalization Across Different Views on One Dataset (RQ3)

Firstly, we investigate the effect of MinT on one specific dataset. The GSM8K dataset is selected for this investigation, as it is annotated by the $\mathrm{CoT}_{\mathrm{clean}}$ view wherein the equations are enclosed by $<<$ and $>>$, and thus $\mathrm{CoT}_{\mathrm{clean}}$ can be relatively easily transformed into EQN and Tree views, offering a suitable platform for our investigation. We first consider the baselines that are fine-tuned on GSM8K using each of the three views individually. Then, we introduce a model fine-tuned with our proposed approach and evaluate it on different views by postpending view-specific instructions to questions.

Prior Work that Use LLMs		**Accuracy**
(Shridhar et al., 2022) (GPT-6B)		21.0%
(Fu et al., 2023) (FlanT5-11B)		27.1%
(Magister et al., 2022) (T5-11B)		38.2%
Multi-Vi**e**w Fi**n**e-**T**uning (MinT 🌿)		
Train View	**Test View**	**Accuracy**
$\mathrm{CoT}_{\mathrm{clean}}$	$\mathrm{CoT}_{\mathrm{clean}}$	35.4%
EQN	EQN	30.5%
Tree	Tree	32.3%
$\mathrm{CoT}_{\mathrm{clean}}$+EQN	$\mathrm{CoT}_{\mathrm{clean}}$	35.9%
$\mathrm{CoT}_{\mathrm{clean}}$+EQN	EQN	36.2%
$\mathrm{CoT}_{\mathrm{clean}}$+EQN+Tree	$\mathrm{CoT}_{\mathrm{clean}}$	36.5%
$\mathrm{CoT}_{\mathrm{clean}}$+EQN+Tree	EQN	36.9%
$\mathrm{CoT}_{\mathrm{clean}}$+EQN+Tree	Tree	**37.8%**

Table 4: Results on different views of GSM8K.

Table 4 shows that augmenting the data with additional views improves performance on all views. Fine-tuning on the original $\mathrm{CoT}_{\mathrm{clean}}$ annotations achieves 35.4% accuracy on the test set ($\mathrm{CoT}_{\mathrm{clean}}$ view). Adding EQN and Tree views during training boosts $\mathrm{CoT}_{\mathrm{clean}}$ accuracy to 36.5%, a 1.1% absolute improvement. More substantial gains are observed on the Tree view (from 32.3% to 37.8%).

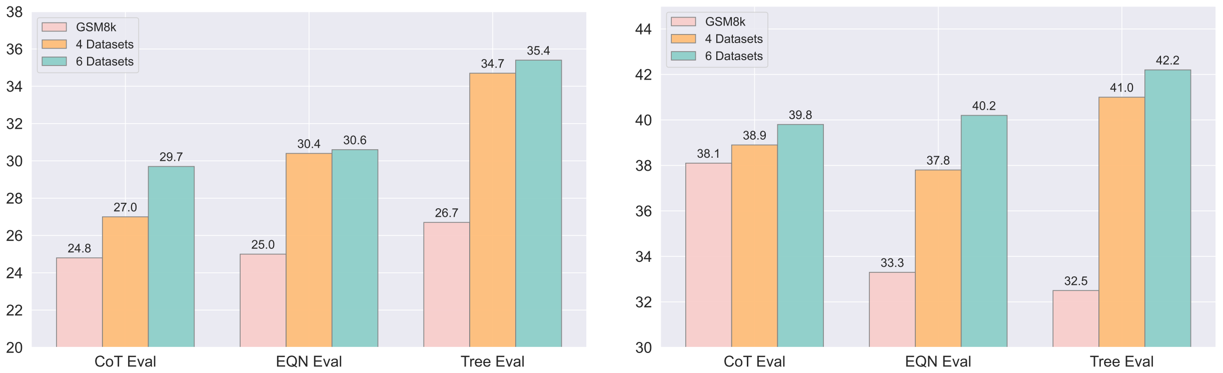
Also, we note that prior work utilizes high-quality data generated by LLMs and different model architectures. As such, we would like to clarify that our work is orthogonal to them and thus, we are not intending to make direct comparisons, though including them for reference may be useful. In fact, our MinT can seamlessly combine with those data-centric methods, by simply assigning additional views for the generated data from stronger models as shown in Section 4.3.

4.5. Evaluation on Held-out Dataset

In order to further assess the multi-view problem solving abilities of our method, we evaluated it on the held-out dataset, MAWPS (Koncel-Kedziorski et al., 2016), which contains 2,373 English MWPs annotated with Equation Solutions (ES) view. It integrates several datasets (Hosseini et al., 2014; Kushman et al., 2014; Koncel-Kedziorski et al., 2015; Roy and Roth, 2015) and thus serves as a comprehensive benchmark. Three training data settings are used: GSM8K only, four clean datasets, and all six datasets, where respective models are all trained with MinT. Our results in Figure 3 are similar to the observations from our previous experiments:

	GSM8K			MathQA		CM17K	Ape210K	
Simple Dataset Mixture	35.3			81.0		68.9	74.1	
a1. Original View with Instructions	35.5			80.2		76.6	74.7	
a2. Only EQN View with Instructions	34.6			79.9		77.5	74.0	
a3. Only Tree View with Instructions	39.6			81.0		-	75.2	
MinT (Our method)	$\text{CoT}_{\text{clean}}$	EQN	Tree	EQN	Tree	EQN	EQN	Tree
	38.8	39.2	40.8	81.0	81.3	77.6	76.0	74.3

Table 5: We use GSM8K, MathQA, CM17K, and Ape210K in our ablation study with different training strategies described in 4.6 in *Italics*. The first ablation (a1) aims to investigate the impact of the instructions, while the rest two ablations (a2, a3) can examine the improvement brought by the generalization from other views.



(a) Results on BLOOMz-7B.

(b) Results on Vicuna-7B.

Figure 2: Experimental results with different backbones on GSM8K. X-axis indicates the evaluation views and Y-axis indicates the accuracy.

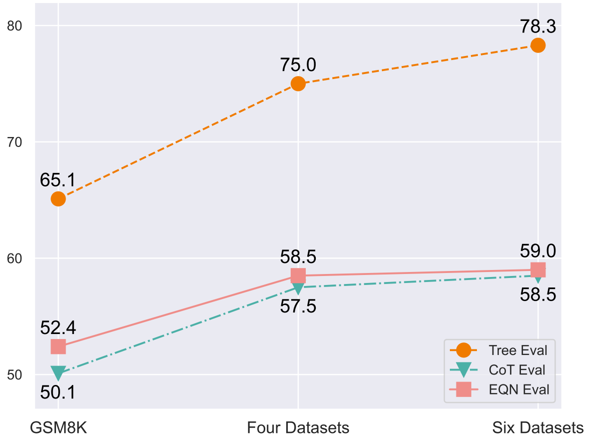


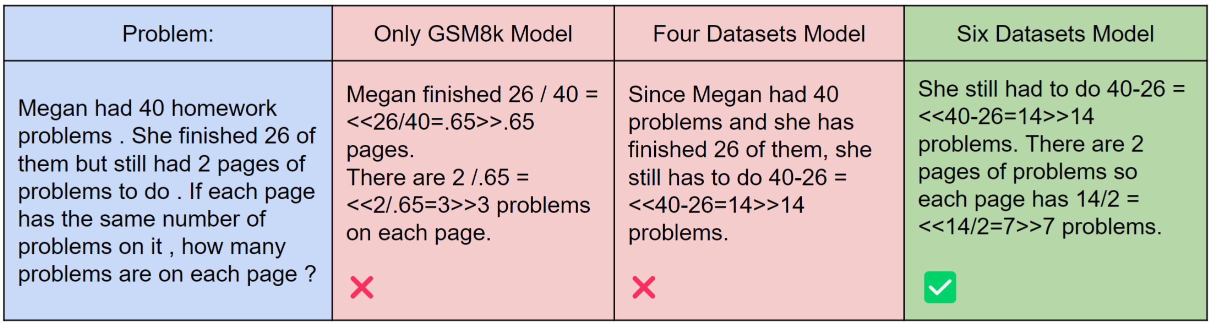
Figure 3: Experimental results on the MAWPS dataset. X-axis indicates the training datasets and Y-axis indicates the accuracy.

increasing the number of datasets used in training with MinT boosts performance across all views. It is noteworthy that although the MAWPS dataset is originally annotated using the EQN view, our model manages to attain an accuracy of 58.5% when attempting to solve problems using step-by-step CoTs. This observation indicates that the problem-solving ability acquired from the training datasets can indeed be transferred to the held-out datasets. More interestingly, it suggests that our method may be used to facilitate tasks like held-out data annotation.

4.6. Ablation Study

To demonstrate the effectiveness of our proposed approach. We implement three ablated studies: *1) Only use the original view for every dataset, and postpending the corresponding instructions to the training samples; 2) Only use the EQN view from GSM8K, MathQA, CM17K, and Ape210K and keep the instructions; 3) Only use the Tree view from GSM8K, MathQA and Ape210K and also keep the instructions.*

As shown in Table 5, the first ablation proves that the instructions can bring some improvements on certain datasets. And the other two ablations show that unifying the views of solution can also bring some improvements to the performance. Nonetheless, when compared with these baselines and ablations, our proposed MinT performs the best, thereby reaffirming that MinT truly capitalizes on the generalization across diverse solution views of mathematical problems.



Problem:	Only GSM8k Model	Four Datasets Model	Six Datasets Model
Megan had 40 homework problems . She finished 26 of them but still had 2 pages of problems to do . If each page has the same number of problems on it , how many problems are on each page ?	Megan finished 26 / 40 = <<26/40=.65>>.65 pages. There are 2 /.65 = <<2/.65=3>>3 problems on each page. ✗	Since Megan had 40 problems and she has finished 26 of them, she still has to do 40-26 = <<40-26=14>>14 problems. ✗	She still had to do 40-26 = <<40-26=14>>14 problems. There are 2 pages of problems so each page has 14/2 = <<14/2=7>>7 problems. ✓

Figure 4: Our case study on the held-out dataset - MAWPS.

4.7. Adaptivity on Different Backbones

We replace our original backbone model LLaMA-7B with other two state-of-the-art models, namely BLOOMz-7B (Muennighoff et al., 2022) and Vicuna-7B (Chiang et al., 2023). This substitution allows us to observe how well MinT adapts to different language models. We trained three different models on them: one trained on the GSM8K dataset, another trained on a combination of four clean math datasets, and a third trained on a total of six datasets that additionally consider noisy data.

4.8. Case Study

Figure 4 illustrates a sample problem from the MAWPS dataset, which is fed into three distinct models discussed in Section 4.4. The first model yields an completely wrong solution, possibly due to the significant disparities in solution patterns between GSM8K and MAWPS, leading to the model's inability to generalize for problems in the held-out set. The second model's solution, though only partially correct and incomplete, suggests an improvement in its reasoning capabilities. The third model, trained with all six datasets, effectively solves the problem in this example, thereby affirming the efficacy of our methodology in improving mathematical reasoning ability.

As illustrated in Figure 2, our method demonstrates the same pattern on both backbones as that on the LLaMA-7B backbone, i.e., incorporating more training data can further enhance performance with the aid of MinT. Also, it is notable that the Vicuna backbone has a better performance on the $\mathrm{CoT}_{\mathrm{clean}}$ view. This may be due to that the Vicuna model is more familiar with the "explanation" style data than symbolic equations as it is continually fine-tuned on dialogues. This means that the effectiveness of our method is not restricted to one specific model and it can also be extended to other LMs and benefits from their own characters. This consistent performance across different backbones validates the robustness of MinT and supports its applicability in a wider range.

5. Discussion

5.1. Conclusions

In this paper, we propose MinT, a novel multi-view fine-tuning approach to enhance the mathematical reasoning capabilities of language models. By framing diverse annotation formats across datasets as distinct "view" of solutions, our method enables models to learn from these unique problem-solving perspectives. We also provide a data-efficient view-transformation strategy, expanding the model's ability to generalize and reason by converting the data annotated in one view into multiple different views. Through MinT, our model exhibits strong performance on multiple benchmarks, outperforming prior knowledge distillation-based techniques. Our experiments demonstrate promising generalization ability and adaptivity to noisy data, held-out data, and across model architectures.

5.2. Broader Impact

We believe MinT provides a scalable and flexible approach for specialized LMs by supervised fine-tuning. Many other reasoning tasks, such as commonsense or symbolic reasoning, can be solved through diverse paths. Investigating how to leverage MinT for general flexible reasoning is an exciting future direction.

Furthermore, MinT demonstrates effective control over language model fine-tuning. By guiding the model with simple instruction strings, we can take advantage of different types and even incomplete and irrelevant data, while still performing well for downstream tasks. This opens possibilities for future design of large-scale general instruction tuning and task-specific fine-tuning.

Additionally, our approach could be seamlessly integrated with verifier-based methods (Cobbe et al., 2021; Zhu et al., 2023; Lightman et al., 2023), reinforcement learning (Ouyang et al., 2022; Touvron et al., 2023b) and rejection sampling (Yuan et al., 2023). Using multi-view verifiers or reward models to score model outputs could provide stronger feedback signals to guide the training/fine-tuning of language models.

6. Bibliographical References

Reem Alghamdi, Zhenwen Liang, and Xiangliang Zhang. 2022. Armath: a dataset for solving arabic math word problems. In *LREC*, pages 351–362.

Zeyuan Allen-Zhu and Yuanzhi Li. 2023. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In *ICLR*.

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 *NAACL-HLT*, pages 2357–2367.

Steffen Bickel and Tobias Scheffer. 2004. Multi-view clustering. In *ICDM*, volume 4, pages 19–26. Citeseer.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *NeurIPS*, 33:1877–1901.

Xiaochun Cao, Changqing Zhang, Huazhu Fu, Si Liu, and Hua Zhang. 2015. Diversity-induced multi-view subspace clustering. In *CVPR*, pages 586–594.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. *ICML*.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2022. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*.

Yining Hong, Qing Li, Daniel Ciao, Siyuan Huang, and Song-Chun Zhu. 2021. Learning by fixing: Solving math word problems with weak supervision. In *AAAI*, volume 35, pages 4959–4967.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *EMNLP*, pages 523–533.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander 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. *arXiv preprint arXiv:2305.02301*.

Zhanming Jie, Jierui Li, and Wei Lu. 2022. Learning to reason deductively: Math word problem solving as complex relation extraction. In *ACL*, pages 5944–5955.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *NeurIPS*, 35:22199–22213.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. 2015. Parsing algebraic word problems into equations. *TACL*, 3:585–597.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. Mawps: A math word problem repository. In *NAACL*, pages 1152–1157.

Nate Kushman, Luke Zettlemoyer, Regina Barzilay, and Yoav Artzi. 2014. Learning to automatically solve algebra word problems. In *ACL*, pages 271–281.

Yeqing Li, Feiping Nie, Heng Huang, and Junzhou Huang. 2015. Large-scale multi-view spectral clustering via bipartite graph. In *AAAI*, volume 29.

Zhongli Li, Wenxuan Zhang, Chao Yan, Qingyu Zhou, Chao Li, Hongzhi Liu, and Yunbo Cao. 2022. Seeking patterns, not just memorizing procedures: Contrastive learning for solving math word problems. In *Findings of ACL*, pages 2486–2496.

Zhenwen Liang, Kehan Guo, Gang Liu, Taicheng Guo, Yujun Zhou, Tianyu Yang, Jiajun Jiao, Renjie Pi, Jipeng Zhang, and Xiangliang Zhang. 2024. Scemqa: A scientific college entrance level multimodal question answering benchmark. *arXiv preprint arXiv:2402.05138*.

Zhenwen Liang, Tianyu Yang, Jipeng Zhang, and Xiangliang Zhang. 2023a. Unimath: A foundational and multimodal mathematical reasoner. In *EMNLP*, pages 7126–7133.

Zhenwen Liang, Wenhao Yu, Tanmay Rajpurohit, Peter Clark, Xiangliang Zhang, and Ashwin Kaylan. 2023b. Let gpt be a math tutor: Teaching math word problem solvers with customized exercise generation. *arXiv preprint arXiv:2305.14386*.

Zhenwen Liang, Jipeng Zhang, Lei Wang, Wei Qin, Yunshi Lan, Jie Shao, and Xiangliang Zhang. 2022a. Mwp-bert: Numeracy-augmented pretraining for math word problem solving. In *Findings of NAACL*, pages 997–1009.

Zhenwen Liang, Jipeng Zhang, Lei Wang, Yan Wang, Jie Shao, and Xiangliang Zhang. 2023c. Generalizing math word problem solvers via solution diversification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13183–13191.

Zhenwen Liang, Jipeng Zhang, and Xiangliang Zhang. 2022b. Analogical math word problems solving with enhanced problem-solution association. In *EMNLP*, pages 9454–9464.

Zhenwen Liang, Jipeng Zhang, and Xiangliang Zhang. 2023d. Don't be blind to questions: Question-oriented math word problem solving. In *AACL*, pages 15–25.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Qian Liu, Fan Zhou, Zhengbao Jiang, Longxu Dou, and Min Lin. 2023. From zero to hero: Examining the power of symbolic tasks in instruction tuning. *arXiv preprint arXiv:2304.07995*.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2022. Teaching small language models to reason. *arXiv preprint arXiv:2212.08410*.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing english math word problem solvers. In *ACL*, pages 975–984.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Kamal Nigam and Rayid Ghani. 2000. Analyzing the effectiveness and applicability of co-training. In *CIKM*, pages 86–93.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv*.

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

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? In *NAACL*, pages 2080–2094.

Jinghui Qin, Xiaodan Liang, Yining Hong, Jianheng Tang, and Liang Lin. 2021. Neural-symbolic solver for math word problems with auxiliary tasks. In *ACL*, pages 5870–5881.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *EMNLP*, pages 1743–1752.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *ICLR*.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2022. Distilling multi-step reasoning capabilities of large language models into smaller models via semantic decompositions. *arXiv preprint arXiv:2212.00193*.

Linfeng Song, Ante Wang, Jinsong Su, Yue Zhang, Kun Xu, Yubin Ge, and Dong Yu. 2020. Structural information preserving for graph-to-text generation. In *ACL*, pages 7987–7998.

Shiliang Sun and Feng Jin. 2011. Robust co-training. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(07):1113–1126.

Anaïs Tack and Chris Piech. 2022. The ai teacher test: Measuring the pedagogical ability of blender and gpt-3 in educational dialogues. In *Proceedings of the 15th International Conference on Educational Data Mining*, page 522.

Minghuan Tan, Lei Wang, Lingxiao Jiang, and Jing Jiang. 2022. Investigating math word problems using pretrained multilingual language models. *MathNLP 2022*, page 7.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lei Wang, Yan Wang, Deng Cai, Dongxiang Zhang, and Xiaojiang Liu. 2018. Translating a math word problem to a expression tree. In *EMNLP*, pages 1064–1069.

Rui Wang, Junyi Ao, Long Zhou, Shujie Liu, Zhihua Wei, Tom Ko, Qing Li, and Yu Zhang. 2022. Multi-view self-attention based transformer for speaker recognition. In *ICASSP*, pages 6732–6736. IEEE.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 35:24824–24837.

Zhipeng Xie and Shichao Sun. 2019. A goal-driven tree-structured neural model for math word problems. In *IJCAI*, pages 5299–5305.

Dian Yu, Kai Sun, Dong Yu, and Claire Cardie. 2021. Self-teaching machines to read and comprehend with large-scale multi-subject question-answering data. In *Findings of EMNLP*, pages 56–68.

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

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*.

Jipeng Zhang, Lei Wang, Roy Ka-Wei Lee, Yi Bin, Yan Wang, Jie Shao, and Ee-Peng Lim. 2020. Graph-to-tree learning for solving math word problems. In *ACL*, pages 3928–3937.

Wei Zhao, Mingyue Shang, Yang Liu, Liang Wang, and Jingming Liu. 2020. Ape210k: A large-scale and template-rich dataset of math word problems. *arXiv preprint arXiv:2009.11506*.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, Hua Jin, and Dacheng Tao. 2023. Knowledge graph augmented network towards multiview representation learning for aspect-based sentiment analysis. *TKDE*.

Xinyu Zhu, Junjie Wang, Lin Zhang, Yuxiang Zhang, Ruyi Gan, Jiaxing Zhang, and Yujiu Yang. 2023. Solving math word problem via cooperative reasoning induced language models. *ACL*.