Logits-Based Finetuning

Jingyao Li[1], Senqiao Yang[1], Sitong Wu[1], Han Shi[2], Chuanyang Zheng[1], Hong Xu[1], Jiaya Jia[3]

[1]The Chinese University of Hong Kong
[2]Huawei Noah's Ark Lab
[3]Hong Kong University of Science and Technology

Abstract

In recent years, developing compact and efficient large language models (LLMs) has emerged as a thriving area of research. Traditional Supervised Fine-Tuning (SFT), which relies on singular ground truth labels, often fails to capture token-level dependencies and linguistic diversity. To address these limitations, we propose a logits-based fine-tuning framework that integrates the strengths of supervised learning and knowledge distillation. Our approach constructs enriched training targets by combining teacher logits with ground truth labels, preserving both correctness and linguistic diversity. This ensures more reliable and effective training. We constructed a large-scale 1.2M logits dataset and trained a series of science-focused models. Experimental results demonstrate that our method achieves significant improvements, with accuracy gains of 18% on Mawps and 22.7% on TabMWP. Across nine widely used mathematical benchmarks, our method consistently outperforms prior SFT models, achieving an average improvement of 7.28%. Codes are available at https://github.com/dvlab-research/Logits-Based-Finetuning.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across a wide range of NLP tasks (Brown et al., 2020; Thoppilan et al., 2022; Chowdhery et al., 2022; OpenAI, 2023; Anil et al., 2023), yet their immense computational demands pose significant challenges for deployment in resource-constrained environments.

To address this, researchers have focused on developing compact and efficient LLMs, with Supervised Fine-Tuning (SFT) as a widely adopted approach. However, SFT suffers from inherent limitations, particularly its inability to capture inter-token relationships and linguistic diversity. For instance, as illustrated in Fig. 2, multiple valid expressions of the same idea, such as "There are 12 inches in 1 foot" and "There are 12 inches in each foot," highlight the nuanced token-level dependencies that SFT often overlooks. This limitation stems from SFT's reliance on singular ground truth labels or teacher outputs, which fail to account for the richness of alternative phrasings. Consequently, the benefits of SFT are constrained by its inability to fully exploit the intrinsic relationships between tokens.

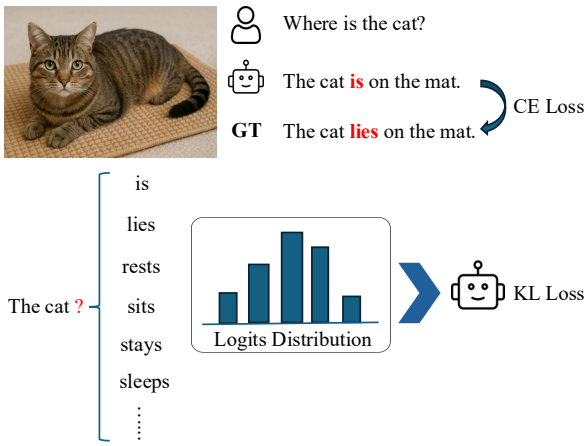

Figure 1: Conceptual overview of our logits-based distillation framework. (Up) Traditional supervised fine-tuning relies on singular ground truth labels, failing to capture valid linguistic variations (e.g., "The cat is on the mat" vs. "The cat lies on the mat"). (Down) Our approach combines teacher model logits with ground truth verification to create enriched training targets that preserve both correctness and expression diversity.

Distillation methods have proven successful in creating lightweight and efficient models. For example, models like BERT (Rusu et al., 2015; Sanh et al., 2019; Jianping et al., 2021) have demonstrated that distillation-based approaches can achieve superior performance compared to direct training methods, offering both efficiency and effectiveness. However, applying distillation to LLMs presents unique challenges. First, the uncontrollability of teacher outputs poses a significant

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 14752–14764
November 4-9, 2025

hurdle. Even well-trained large language models, such as LLaMA3.1-70B-instruct, can generate hallucinated or erroneous predictions, as shown in Tab. 1. Relying solely on such outputs as supervision signals is unreliable and often necessitates manual intervention to ensure high-quality annotations. Second, the computational cost of large-scale distillation is prohibitive, as LLMs require substantial GPU memory, making direct online teacher-student distillation impractical for many applications.

To address these challenges, we propose a novel logits-based fine-tuning framework that integrates the strengths of supervised learning and knowledge distillation. Our approach constructs enriched training targets by combining teacher logits with ground truth labels, preserving both correctness and linguistic diversity. Unlike traditional distillation methods, which transfer teacher predictions directly, our method creates a balanced target distribution that enhances the student model's ability to learn from both the teacher's knowledge and task-specific supervision. This ensures more reliable and informed training while mitigating the risks associated with erroneous teacher outputs.

In this work, we constructed a large-scale 1.2M logits dataset and trained a series of science-focused models using our method. Experimental results show that our approach surpasses the previous state-of-the-art methods on Mawps and TabMWP by 18% and 22.7% in accuracy, respectively. Across nine widely used mathematical benchmarks, our method consistently outperforms prior SFT models, with an average improvement of 7.28%, highlighting the method's robustness and generalizability.

In summary, the main contributions of our work are as follows:

1. We propose a simple yet effective logits-based instruction tuning method that enhances model performance by integrating teacher knowledge with ground truth labels.
2. We release a 1.2M science logits dataset, enabling future research and development of logits-based training methods.
3. We train and evaluate a series of science-focused models using our method. Our models achieve significant improvements over state-of-the-art supervised fine-tuning approaches, with an average accuracy gain of 7.28% across nine benchmarks.

2 Preliminaries

In this section, we establish the theoretical foundation for our logits-based fine-tuning approach. We first formalize auto-regressive sequence modeling and then analyze existing knowledge distillation paradigms, highlighting their limitations that motivate our method.

2.1 Auto-regressive Sequence Models

We first define key components of sequence modeling. For any sequence pair, x represents the input and y the output. The vocabulary $\mathbb{V}$ contains M distinct tokens. We use $y_{<n+1} = (y_1, y_2, \ldots, y_n)$ to represent the first n tokens. An auto-regressive model generates a probability distribution $p(.|y_{<n}, x) \in [0,1]^M$ over the vocabulary $\mathbb{V}$, considering both input x and previous tokens $y_{<n}$. When sampling, $y \sim p(\cdot|x)$ produces a complete output sequence. For brevity, we write $p(y_n|x)$ instead of $p(y_n|y_{<n}, x)$. The generation process predicts tokens sequentially. Each token probability $p(y_n|x)$ is computed using a temperature-controlled softmax:

$$p(y_n|x) = \frac{e^{z_n/\tau}}{\sum_{i=1}^{M} e^{z_i/\tau}}, \quad (1)$$

where z_n represents the logit for token y_n, and τ controls output randomness. Higher τ increases diversity, while lower values produce more focused predictions. During student training, $\tau = 1$, while evaluation uses greedy sampling ($\tau \to 0$).

2.2 Distillation for Auto-regressive Models

KL-Based Divergences. The Kullback-Leibler (KL) divergence is a fundamental measure that quantifies the difference between two probability distributions. For two discrete probability distributions $P(\cdot|x)$ and $Q(\cdot|x)$ defined over the probability space X, the KL divergence is defined as (Hinton et al., 2015):

$$\mathcal{D}_{KL}(P|Q) = \sum_{x \in \mathcal{X}} P(\cdot|x) \log \frac{P(\cdot|x)}{Q(\cdot|x)}. \quad (2)$$

This measure is always non-negative and equals zero if and only if the two distributions are identical.

Supervised FT. Given a fixed dataset of target sequences, one simple strategy involves maximizing the student's negative log-likelihood on these sequences:

$$L_{SFT}(\theta) = \mathbb{E}_{(x,y)\sim(X,Y)}\big[-\log p_S^{\theta}(y|x)\big]. \quad (3)$$

Instruction

Roger's age is 5 more than twice Jill's age. In 15 years, their age difference will be 30 years less than Finley's age. If Jill is 20 years old now, how old is Finley?

Teacher Annotations	**Human Annotations**
Let's break down the problem step by step. First, we know that Jill is 20 years old now. We also know that Roger's age is 5 more than twice Jill's age. So, Roger's age is $2 \times 20 + 5 = 45$. In 15 years, Jill will be 20 + 15 = 35 years old, and Roger will be 45 + 15 = 60 years old. The problem states that their age difference will be 30 years less than Finley's age. So, we can set up the equation: 60 - 35 = Finley's age - 30 Simplifying the equation, we get: 25 = Finley's age - 30 Adding 30 to both sides, we get: Finley's age = 55 Therefore, Finley is **55** years old now.	Twice Jill's age is $2 \times 20 = 40$. If Roger's age is 5 more than twice Jill's age, he is 5+40=45 years old. In 15 years, Jill will be 20+15=35 years old. Roger's age will be 45+15=60 years old in 15 years. The age difference between Roger and Jill in fifteen years will be 60-35=25 years. In 15 years, the age difference between Roger and Jill will be 30 years less than Finley's age, meaning in fifteen years Finley will be 30+25=55 years old. Currently, Finley is **55-15=40** years old.

Table 1: An example from GSM8K (Cobbe et al., 2021) generated by teacher model, LLaMA3.1-70B-instruct. The teacher model incorrectly interprets Finley's age 15 years from now as his current age, leading to an erroneous conclusion.

Sequence-Level KD (Kim and Rush, 2016a) extends this concept by training on teacher-generated outputs Y_T. This approach optimizes:

$$L_{SeqKD}(\theta) = \mathbb{E}_{(x,y)\sim(X,Y_T)}\big[-\log p_{\mathrm{S}}^{\theta}(y|x)\big]. \quad (4)$$

Supervised KD (Hinton et al., 2015) represents a widely used distillation method where students learn to match their teacher's token-level probability distributions. The training objective minimizes the KL divergence between teacher and student distributions:

$$L_{SD}(\theta) := \mathbb{E}_{(x,y)\sim(X,Y_T)}\Big[\mathcal{D}_{KL}\big(p_{\mathrm{T}}|p_{\mathrm{S}}^{\theta}\big)(y|x)\Big], \quad (5)$$

3 Logits-based Finetuning

In this section, we first introduce the motivation behind our logits-based fine-tuning approach in Sec. 3.1. Then, in Sec. 3.2, we present the proposed distribution, which integrates teacher model logits with ground truth outputs. In Sec. 3.3, we describe the construction of our logits dataset. Finally, in Sec. 3.4, we detail our fine-tuning method.

3.1 Motivation

To justify the proposal of the Logits-Based Fine-Tuning method for improving small LLMs, we first analyze the limitations of traditional widely used method Supervised Fine-Tuning (SFT), and the current distillation method Sequence-Level Knowledge Distillation (SeqKD, Kim and Rush (2016a)), and Supervised Distillation (SD, Hinton et al. (2015)):

Lack of Inter-Token Relationships. For traditional SFT, the major issue is the lack of inter-token relationships. Specifically, there may be multiple expressions for the same idea, such as *There are 12 inches in 1 foot* and *There are 12 inches in each foot* illustrated in Fig. 2. These alternative labels reflect the model's understanding of the intrinsic relationships between tokens, which may not be captured through singular annotations.

Uncontrollability of Teacher Outputs. Besides, for the distillation method, the outputs from LLMs are often uncontrollable; even well-trained models can produce erroneous or hallucinatory results. For instance, as shown in Tab. 1, the well-trained LLaMA3.1-70B-instruct model erroneously interprets Finley's age 15 years from now as his current age, resulting in incorrect conclusions. Therefore, relying solely on the outputs of LLMs as supervision for models is unreliable and necessitates human intervention to generate validated results.

3.2 Target Distribution Analysis

To address these limitations, we aim to propose a approach that enables the student model to learn from both reliably annotated labels and the intrinsic knowledge embedded in the teacher model.

Problem Setup. Consider two sequence models with auto-regressive architectures: p_{S} (student) and

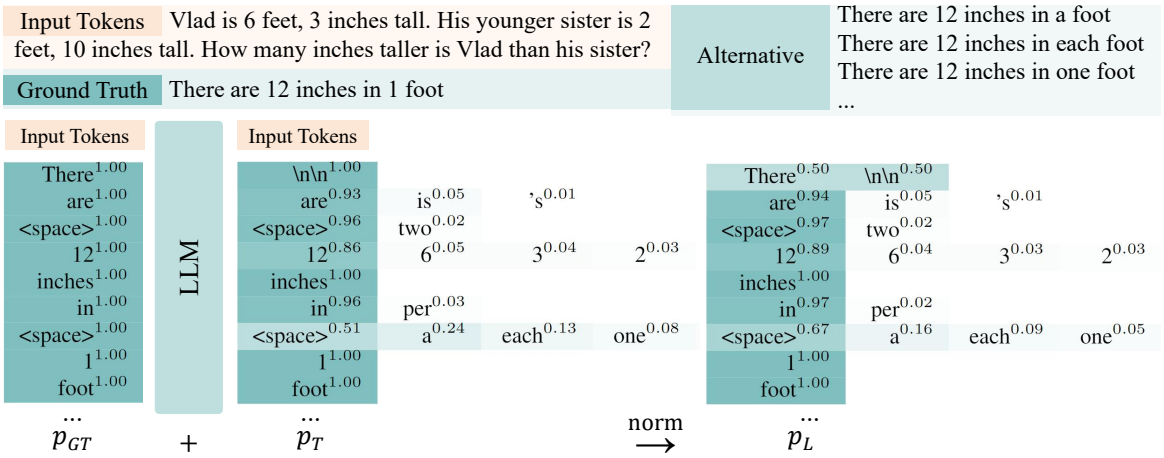

Figure 2: Illustration of token probability distribution generation. The input tokens concated with ground truth are processed by the teacher model, which predicts the next token probabilities p_T. Then the ground truth one-hot vector P_{GT} is combined with the teacher's top-K probabilities p_T to generate the proposed distribution p_{our} using Eq. (7).

p_{T} (teacher), with different model capacities. The student model has trainable parameters θ, and p_{S}^{θ} maintains differentiability with respect to θ. The setup includes an input dataset X. We define the token-level distribution discrepancy between p_{T} and p_{S} as:

$$\begin{aligned}&\mathcal{D}\big(p_{\text{T}}\|p_{\text{S}}^{\theta}\big)(y|x)\\&:=\frac{1}{L_y}\sum_{n=1}^{L_y}\mathcal{D}\big(p_{\text{T}}(\cdot|y_{<n},x)\|p_{\text{S}}^{\theta}(\cdot|y_{<n},x)\big),\end{aligned}\tag{6}$$

where x and y denote the input and output sequences and $\mathcal{D}$ represents divergence measure.

Definition. Let M represent the vocabulary size and y_i denote the i-th ground truth index, where $0 < y_i < M$. The target distribution is denoted as q. Specifically, $q_j(y_i)$ represents the value at the j-th position in the vocabulary for the i-th token's logits in the target logits q. Storing a vocabulary of millions of tokens incurs significant storage overhead. Therefore, we retain only the sparse teacher logits of the top K instead of the complete set. For simplicity, all subsequent references to p_{T} logits refer to the Top-K sparsified results. We define $\text{Top}_K p_{\text{T}}(y_i) = \text{Top}_{K,1\le j\le M} p_{\text{T}}(y_i)$.

Proposed Distribution. We propose our probability distribution p_{L} as follows:

$$p_{\text{L}}(y_i) = \frac{p_{\text{T}}(y_i) + p_{\text{GT}}(y_i)}{\|p_{\text{T}}(y_i) + p_{\text{GT}}(y_i)\|_1},\tag{7}$$

where $\|\cdot\|_1$ denotes the L1 norm. $p_{\text{GT}}(y_i)$ is the one-hot encoded ground truth label. Specifically, $p_{\text{GT}}(y_i) = \{p_{\text{GT}j}(y_i)\}_{j=1}^{M} \in [0,1]^M$, where

$$p_{\text{GT}}(y_i)^j = \begin{cases}1, & \text{if} j = y_i,\\ 0, & \text{otherwise.}\end{cases}\tag{8}$$

We define this distribution because it satisfies the following constraints.

Constraint 1. To ensure that the greedy search on the new distribution q still yields the ground truth y_i, we require that the value $q(y_i)$ be the largest at the ground truth index. Mathematically, this is expressed as:

$$q_{y_i}(y_i) \ge q_j(y_i), \quad \forall 1 \le j \le M, j \ne y_i \tag{9}$$

This constraint guarantees that the argmax of $q(y_i)$ remains y_i, preserving the ground truth prediction.

Constraint 2. We aim to maintain the relative proportions of the top K candidates from the original distribution $p_{\text{T}}(y_i)$ in the new distribution q. The constraint is formulated as:

$$\frac{q_j(y_i)}{q_k(y_i)} = \frac{p_{\text{T}}(y_i)_j}{p_{\text{T}}(y_i)_k}, \quad \forall j,k \in \text{Top}_K(y_i), j,k \ne y_i \tag{10}$$

This ensures that the proportional relationship between the probabilities determined by the original distribution is preserved in the new distribution.

Constraint 3. For indices outside the ground truth and the top K candidates, we require their values in q to be not larger than those within the set $S = \{y_i\} \cup \text{Top}_K p_{\text{T}}(y_i)$. This is expressed as:

$$q_j(y_i) \le q_k(y_i), \quad \forall j \notin S, \forall k \in S \tag{11}$$

Algorithm 1 Logits Dataset Generation Procedure

Require: Teacher model p_{T}, Dataset $(X, Y) = (x_i, y_i,)_{i=1}^{N}$
Ensure: Logits-based Dataset $(X, Y, P_{\mathrm{L}}) = (x_i, y_i, p_{\mathrm{L}i})_{i=1}^{N}$

1: **for** each $(x, y) \in (X, Y)$ **do**
2: Compute Top-K teacher logits $p_{\mathrm{T}} \leftarrow T(x)$
3: Create one-hot ground truth p_{GT} using Eq. (8).
4: Compute p_{L} using Eq. (7).
5: **end for**
6: **return** Logits-based Dataset $(X, Y, P_{\mathrm{L}}) = (x_i, y_i, p_{\mathrm{L}i})_{i=1}^{N}$

Algorithm 2 Logits-based Finetuning Procedure

Require: Student model p_{S}^{θ}, Logits-based Dataset (X, Y, P_{L}), Divergence $\mathcal{D}$, learning rate η
Ensure: Trained student model p_{S}^{θ}

1: **for** batch $B \in (X, Y, P_{\mathrm{L}})$ **do**
2: Update student parameters θ by minimizing L_{L} (Eq. (13)):

$$\theta \leftarrow \theta - \eta \frac{1}{B} \sum_{(x,y,p_{\mathrm{L}}) \in B} \nabla_{\theta} \mathcal{D}(p_{\mathrm{L}} \| p_{\mathrm{S}}^{\theta})(y|x)$$

3: **end for**
4: **return** Trained student model p_{S}^{θ}

This constraint helps in focusing the probability mass on the ground truth and the top candidates, reducing the influence of less relevant tokens.

Constraint 4. Finally, the new distribution $q(y_i)$ must be a valid probability distribution. This implies that each element must be within the range [0, 1], and the sum of all elements must equal 1. Mathematically:

$$q(y_i) \in [0, 1]^M, \sum_{j=1}^{M} q_j(y_i) = 1 \qquad (12)$$

These constraints ensure that $q(y_i)$ is a well-formed probability distribution, suitable for logits-based fine-tuning. It can be easily demonstrated that p_{L} satisfies the four constraints outlined above. Details are in Sec. A.

3.3 Logits Dataset Generation

The logits dataset generation procedure, as detailed in Alg. 1, takes a standard dataset of input-target pairs and enriches it with target distributions derived from a pre-trained teacher model.

For each input-target pair (x, y), the teacher model p_{T} is first used to compute the full logits vector for input x, which is then sparsified by retaining only the top K logits, denoted as $p_{\mathrm{T}}(x)$. This sparsification is crucial for reducing storage requirements and focusing on the teacher's most confident predictions. Concurrently, a one-hot vector $p_{\mathrm{GT}}(y)$ is created based on the ground truth label y, as defined in Equation 8. The final target distribution $p_{\mathrm{L}}(y)$ is then computed using Equation 7, which combines the sparsified teacher logits $p_{\mathrm{T}}(x)$ and the one-hot ground truth vector $p_{\mathrm{GT}}(y)$. This combination balances the teacher's knowledge with the emphasis on the correct target label. The resulting logits-based dataset (X, Y, P_{L}) is then used to fine-tune a student model, leveraging the target distributions for improved knowledge transfer.

Dataset Details. Table 2 presents the results of supervised fine-tuning of LLaMA3.2-1B-Instruct on a variety of mathematical reasoning datasets, including Socratic (Yue et al., 2024), Stack-Exchange (Yue et al., 2024), Camel-AI (Li et al., 2023), MathInstruct (Jiang et al., 2024), GSM8K (Cobbe et al., 2021), MetaMath (Yu et al., 2023), MetaMath-GSM8K (Yu et al., 2023), and OpenMathInstruct2 (Toshniwal et al., 2024). Among them, OpenMathInstruct2 demonstrates the strongest overall performance, achieving the highest average score (24.6%) and outperforming other datasets on most dataset yields competitive performance (23.8%) and the best result on the Olympiad Bench. These results suggest that datasets like MetaMath-GSM8K and OpenMathInstruct2 can lead to more robust and generalizable mathematical reasoning capabilities. Therefore, our final 1.2M logits dataset consists of 1M samples from MetaMath-GSM8K and 240K from OpenMathInstruct2. More details are shown in Sec. B. The teacher model utilized for logits generation is LLaMA3.1-70B-Instruct (AI@Meta, 2024).

3.4 Finetuning Method

Using the proposed distribution p_L mentioned above, we fine-tune the student model.

Loss Function. Our Logits-based Finetuning (LFT) method uses the Kullback-Leibler (KL) divergence as the loss function to train the student

Dataset	GSM8K	MATH	College Math	GaoKao 2023 en	Minerva Math	Olympiad Bench	Average
Baseline	46.9	31.6	18.6	26.2	5.5	7.0	22.6
Socratic	35.9	20.3	8.8	17.4	3.7	3.4	14.9
ScienceQA	39.7	21.6	11.7	15.8	4.4	5.9	16.5
StackExchange	37.8	22.3	12.9	19.5	3.3	4.6	16.7
Camel-AI	41.0	22.1	11.3	20.3	5.1	3.6	17.2
MathInstruct	40.9	24.4	12.7	20.0	6.2	4.6	18.1
GSM8K	45.7	29.4	16.9	23.6	5.9	5.8	21.2
MetaMath	**54.8**	28.8	21.4	19.7	7.0	7.3	23.2
Metamath-GSM8K	54.1	29.7	21.1	24.9	4.8	**8.1**	23.8
OpenMathInstruct2	49.7	**32.7**	**23.4**	**27.8**	**7.4**	6.7	**24.6**

Table 2: Results of LLaMA3.2-1b-instruct after supervised fine-tuning on various datasets, including Socratic (Yue et al., 2024), StackExchange (Yue et al., 2024), Camel-AI (Li et al., 2023), MathInstruct (Jiang et al., 2024), GSM8K (Cobbe et al., 2021), MetaMath (Yu et al., 2023), MetaMath-GSM8K (Yu et al., 2023), and OpenMathInstruct2 (Toshniwal et al., 2024).

model. The loss function is defined as:

$$L_{\mathrm{L}}(\theta) := \mathbb{E}_{(x,y)\sim(X,Y)}\Big[\mathcal{D}_{KL}\big(p_{\mathrm{L}}|p_{\mathrm{S}}^{\theta}\big)(y|x)\Big], \tag{13}$$

where x and y represent the input and output sequences. (X, Y) is the dataset of input-output pairs. $\mathbb{E}[\cdot]$ denotes the expectation over the dataset.

Fine-tuning. This Logits-Based Fine-tune leverages a pre-generated logits dataset, as described in Sec. 3.3, to guide the training of a student model p_{S}^{θ}. Alg. 2 details our logits-based fine-tuning procedure. For each batch B from the dataset, the student's parameters θ are updated by minimizing the loss L_{L} (Eq. (13)), which measures the divergence $\mathcal{D}$ between p_{L} and $p_{\mathrm{S}}^{\theta}(y|x)$. This process results in a trained student model that incorporates knowledge from the teacher logits and ground truth labels.

4 Experiment

In this section, we present a comprehensive evaluation of our logits-based fine-tuning approach. We first detail our evaluation benchmarks in Sec. 4.1 and training details in Sec. 4.2. Then, we analyze key components through ablation studies in Sec. 4.3. Finally, we compare on multiple datasets against baselines in Sec. 4.4.

4.1 Benchmark

We evaluate our ScienceLLaMA on mathematical benchmark including:

GSM8K (Grade School Math 8K, Cobbe et al. (2021)) is a dataset comprising 8.5K high-quality, linguistically diverse grade school math word problems.

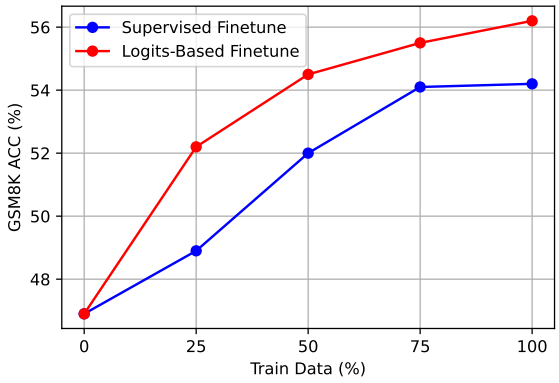

Figure 3: Ablation of our logits-based finetune comparing with baseline trained on different percentage of MetaMath-GSM8K (Yu et al., 2023) and evaluated on GSM8K (Cobbe et al., 2021).

MATH (Hendrycks et al., 2021) consists of 12,500 challenging competition mathematics problems, each accompanied by a detailed step-by-step solution.

OlympiadBench (He et al., 2024) presents an Olympiad-level bilingual multimodal scientific benchmark with 8,476 problems from challenging mathematics and physics competitions like the Chinese college entrance exam.

CollegeMath (Tang et al., 2024) is a mathematical reasoning dataset created using MathScale, containing two million math question-answer pairs.

Data Source	Data Description	Data Count
OpenMathInstruct (Toshniwal et al., 2024)	Problem Synthesize from Math	832k
	Problem Synthesize from GSM8K	138k
	CoT Aug from Math	15k
	CoT Aug from GSM8K	15k
MetaMath (Yu et al., 2023)	Answer Aug from GSM8K	80k
	Rephrasing from GSM8K	80k
	Self-Verification from GSM8K	40k
	Forward-Backward Reasoning from GSM8K	40k

Table 3: Source and description of our 1.2M logits dataset, including 240K from MetaMath-GSM8K (Yu et al., 2023), and 1M from OpenMathInstruct2 (Toshniwal et al., 2024).

	GSM8K
LLaMA3.2-1b-It	46.9
Supervsied Finetune	54.1
Logits-based Finetune	**56.1**

Table 4: Ablation of our logits-based finetune comparing with baseline trained on MetaMath-GSM8K (Yu et al., 2023).

SVAMP (Simple Variations on Arithmetic Math word Problems, Patel et al. (2021)) introduces a challenge dataset for English math word problems.

ASDiv (Academia Sinica Diverse MWP Dataset, Miao et al. (2020)) offers a diverse English math word problem corpus consisting of 2,305 problems,.

MAWPS (MAth Word ProblemS, Koncel-Kedziorski et al. (2016)) is an online repository providing a unified testbed to evaluate algorithms on Math Word Problems.

CarpEN (Computation-intensive AlgebRa Problems, Zhang et al. (2023a)) constructs a Chinese dataset focused on computation-intensive algebra problems.

TabMWP (Tabular Math Word Problems, Lu et al. (2023)) contains 38,431 open-domain grade-level math problems requiring reasoning over textual and tabular data.

4.2 Training Details

We train the LLaMA3.2-1B/3B-Instruct as our model on our constructed 1.2M science logits dataset using our proposed logits-based fine-tuning method. The resulting trained models are referred to as ScienceLLaMA-1B/3B. We set the batch size to 1 and the learning rate to 2×10^{-5}. All experiments are conducted on 8 Nvidia A800 GPUs.

4.3 Ablation

Figure 3 presents the GSM8K accuracy of our logits-based fine-tuning in comparison to supervised fine-tuning, trained on varying percentages of the MetaMath-GSM8K dataset and evaluated on the GSM8K benchmark. Both methods demonstrate improved performance as the proportion of training data increases, but the logits-based fine-tuning consistently outperforms supervised fine-tuning across all data scales. Notably, the accuracy achieved by the logits-based approach with just 25% of the training data exceeds that of the supervised method trained on 50% of the data. Furthermore, with half of the training data, the logits-based approach achieves better results than the supervised method trained on the full dataset. As shown in Tab. 4, on the complete training set, our logits-based fine-tuning achieves an accuracy of 56.1%, surpassing the supervised fine-tuning baseline by 2.0% and outperforming the original pre-trained model by 9.2%. These findings underscore the effectiveness of leveraging logits to guide the learning process.

4.4 Performance

As shown in Tab. 5, we evaluate our proposed method on various math benchmarks. Our ScienceLLaMA significantly outperforms the SFT model. Specifically, the ScienceLLaMA-1B model surpasses the directly SFT-trained LLaMA3.2-1B-Instruct on Mawps and TabMWP by 18% and 22.7% in accuracy, respectively. Furthermore, for the average score across nine benchmarks, our ScienceLLaMA-1B achieves a 7.28% higher accuracy. These results demonstrate that our method

Model	GSM8K	MATH	College Math	Olympiad Bench	Svamp	ASDiv	Mawps	Carp en	TabMWP	Avg
Gemma-2-2b-It	61.9	26.1	20.6	5.3	68.7	77.6	89.7	32.7	42.7	47.26
Phi-3.5-Mini-It	87.2	45.2	35.9	12.3	83.7	85.9	88.1	35.1	55.7	58.79
LLaMA3.2-1b-It	46.9	31.6	18.6	7.0	69.3	70.0	79.3	30.5	33.4	42.96
LLaMA3.2-3b-It	81.3	51.7	34.1	17.2	86.4	89.0	96.7	45.1	70.0	63.50
ScienceLLaMA-1b	55.0	35.1	25.3	7.6	72.5	78.5	87.3	34.8	56.1	50.24
ScienceLLaMA-3b	81.0	51.3	36.3	16.1	88.4	90.6	96.3	46.0	74.3	64.48

Table 5: Performance of our ScienceLLaMA comparing with current SOTAs on various benchmarks, including Socratic (Yue et al., 2024), StackExchange (Yue et al., 2024), Camel-AI (Li et al., 2023), MathInstruct (Jiang et al., 2024), GSM8K (Cobbe et al., 2021), MetaMath (Yu et al., 2023), MetaMath-GSM8K (Yu et al., 2023), and OpenMathInstruct2 (Toshniwal et al., 2024).

exhibits strong stability and generalization, significantly outperforming the Supervised-Finetuning approach.

5 Related Works

Large Language Models. Recently, LLMs have demonstrated remarkable capabilities across a wide range of tasks (Brown et al., 2020; Thoppilan et al., 2022; Chowdhery et al., 2022; OpenAI, 2023; Anil et al., 2023; Yang et al., 2024), including machine translation (Li et al., 2024a), text summarization (Zheng et al., 2024), dialogue generation (Ouyang et al., 2022), and code generation (Li et al., 2024b). While their capacity is impressive, these advanced abilities often emerge only in models with substantial parameter sizes (Kaplan et al., 2020; Wei et al., 2022), which demand significant computational resources. As a result, model compression has become essential to facilitate the practical deployment of LLMs and to support further research in the field.

Knowledge Distillation. Knowledge Distillation (KD; Hinton et al. (2015)), a popular model compression method, focuses on training a smaller student model under the guidance of a larger teacher model (Rusu et al., 2015; Sanh et al., 2019; Jianping et al., 2021). In NLP, KD has been widely applied to classification tasks by replicating the teacher model's output distribution (Song et al., 2020; Liang et al., 2021; Zhang et al., 2023b), internal layer representations (Jiao et al., 2020; Sun et al., 2019), or attention patterns (Wang et al., 2020, 2021). For text generation tasks, traditional KD typically minimizes the Kullback-Leibler divergence (KLD) between the teacher's and student's output distributions, using the teacher's output as supervision at every time step (Sanh et al., 2019) or directly training the student on text sequences generated by the teacher (Kim and Rush, 2016b; Taori et al., 2023; Chiang et al., 2023; Peng et al., 2023). Unlike recent studies (Agarwal et al., 2024; Wen et al., 2023; Ko et al.; Gu et al., 2024), which focus on alternative distribution discrepancy metrics in KD, our work emphasizes the creation of a distribution that integrates the robustness of the ground truth with the teacher's token-level knowledge priors.

6 Conclusion

In this work, we address the limitations of traditional supervised fine-tuning for developing compact and efficient LLMs by introducing a novel logits-based fine-tuning framework. Our approach integrates the strengths of supervised learning and knowledge distillation, constructing enriched training targets that combine teacher logits with ground truth labels. This method preserves both correctness and linguistic diversity, enabling the student model to learn from the teacher's knowledge while maintaining task-specific supervision. We constructed a large-scale 1.2M science logits dataset and trained a series of science-focused models, referred to as ScienceLLaMA. Experimental results demonstrate that our method achieves significant improvements over state-of-the-art supervised fine-tuning approaches, with accuracy gains of 18% on Mawps and 22.7% on TabMWP. Across nine widely used mathematical benchmarks, our method consistently outperforms prior SFT models, achieving an average improvement of 7.28%. These results highlight the robustness of our logits-based fine-tuning framework.

Limitations

While our work successfully introduces a distillation framework tailored for large language models (LLMs) using a logits-based instruction tuning strategy, our experiments were constrained by computational resources, limiting the scale of the evaluated models. We plan to extend this approach to larger model architectures in future work.

Broader Impact

By refining the distillation process to better preserve the teacher model's reasoning capabilities, our method may enable more compact and deployable models. This could make LLM-powered applications—such as real-time conversational assistants, on-device AI tools, and resource-constrained edge computing—more accessible and practical. However, the broader deployment of efficient, distilled models also introduces risks. If misused, malicious actors might exploit distillation techniques to create highly optimized models for harmful purposes, such as generating convincing misinformation or automating fraudulent interactions. Responsible development and rigorous evaluation frameworks will be essential to mitigate these risks while maximizing the societal benefits of our method.

AI Assistance Disclosure

In the preparation of this work, the authors used large language models (LLMs) for writing assistance during manuscript composition. Following initial drafting, the authors reviewed and edited the content as needed and take full responsibility for the final publication.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. *Preprint*, arXiv:2306.13649.

AI@Meta. 2024. Llama 3 model card.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, et al. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*.

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*.

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

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. *Preprint*, arXiv:2306.08543.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *ACL (1)*, pages 3828–3850. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS Datasets and Benchmarks*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max W.F. Ku, Qian Liu, and Wenhu Chen. 2024. Mantis: Interleaved multi-image instruction tuning. *Transactions on Machine Learning Research*, 2024.

Gou Jianping, Yu Baosheng, Stephen J Maybank, and Tao Dacheng. 2021. Knowledge distillation: A survey. *IJCV*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Findings of EMNLP*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Yoon Kim and Alexander M. Rush. 2016a. Sequence-level knowledge distillation. *Preprint*, arXiv:1606.07947.

Yoon Kim and Alexander M Rush. 2016b. Sequence-level knowledge distillation. In *Proceedings of EMNLP*.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. In *Forty-first International Conference on Machine Learning*.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. Mawps: A math word problem repository. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 1152–1157.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for "mind" exploration of large language model society. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Jingyao Li, Pengguang Chen, Sitong Wu, Chuanyang Zheng, Hong Xu, and Jiaya Jia. 2024a. Robocoder: Robotic learning from basic skills to general tasks with large language models. *Preprint*, arXiv:2406.03757.

Jingyao Li, Pengguang Chen, Bin Xia, Hong Xu, and Jiaya Jia. 2024b. Motcoder: Elevating large language models with modular of thought for challenging programming tasks. *Preprint*, arXiv:2312.15960.

Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. 2021. Mix{kd}: Towards efficient distillation of large-scale language models. In *Proceedings of ICLR*.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2023. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *International Conference on Learning Representations (ICLR)*.

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

OpenAI. 2023. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with GPT-4. *arXiv preprint arXiv:2304.03277*.

Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. 2015. Policy distillation. *arXiv preprint arXiv:1511.06295*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Kaitao Song, Hao Sun, Xu Tan, Tao Qin, Jianfeng Lu, Hongzhi Liu, and Tie-Yan Liu. 2020. LightPAFF: A two-stage distillation framework for pre-training and fine-tuning. *arXiv preprint arXiv:2004.12817*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for BERT model compression. In *Proceedings EMNLP*.

Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. 2024. Mathscale: Scaling instruction tuning for mathematical reasoning. In *ICML*. OpenReview.net.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMA model. `https://github.com/tatsu-lab/stanford_alpaca`.

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

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. 2024. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. *Preprint*, arXiv:2410.01560.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of ACL*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of NeurIPS*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. 2023. f-divergence minimization for sequence-level knowledge distillation. In *Proceedings of ACL*.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2024. Visionzip: Longer is better but not necessary in vision language models. *arXiv preprint arXiv:2412.04467*.

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

Xiang Yue, Tuney Zheng, Ge Zhang, and Wenhu Chen. 2024. Mammoth2: Scaling instructions from the web. *Advances in Neural Information Processing Systems*.

Beichen Zhang, Kun Zhou, Xilin Wei, Wayne Xin Zhao, Jing Sha, Shijin Wang, and Ji-Rong Wen. 2023a. Evaluating and improving tool-augmented computation-intensive math reasoning. *arXiv preprint arXiv:2306.02408*.

Rongzhi Zhang, Jiaming Shen, Tianqi Liu, Jialu Liu, Michael Bendersky, Marc Najork, and Chao Zhang. 2023b. Do not blindly imitate the teacher: Using perturbed loss for knowledge distillation. *arXiv preprint arXiv:2305.05010*.

Chuanyang Zheng, Yihang Gao, Han Shi, Minbin Huang, Jingyao Li, Jing Xiong, Xiaozhe Ren, Michael Ng, Xin Jiang, Zhenguo Li, and Yu Li. 2024. Dape: Data-adaptive positional encoding for length extrapolation. *Preprint*, arXiv:2405.14722.

A Verification of Constraints

We now demonstrate that the proposed distribution $p_{\mathrm{L}}(y_i)$ satisfies the four constraints in Sec. 3.2.

Constraint 1. Since $p_{\mathrm{T}}(y_i) \in [0,1]^M$, and $p_{\mathrm{GT}}(y_i)$ is a one-hot vector with a value of 1 at index y_i and 0 elsewhere, the largest value in $p_{\mathrm{T}}(y_i) + p_{\mathrm{GT}}(y_i)$ will always be at index y_i. The normalization by the L1 norm preserves this relationship, ensuring $p_{\mathrm{L}y_i}(y_i) \geq p_{\mathrm{L}j}(y_i)$ for all $j \neq y_i$. Thus, Constraint 1 is satisfied.

Constraint 2. This constraint pertains to the relative proportions within the Top K elements of $p_{\mathrm{T}}(y_i)$. Since $p_{\mathrm{GT}}(y_i)$ only modifies the ground truth index, and the normalization factor is applied uniformly across all elements, the relative proportions among the other Top-K elements remain unchanged. Specifically, for $j, k \in \text{Top-K}(y_i)$ and $j, k \neq y_i$, we have:

$$\begin{aligned}\frac{p_{\mathrm{L}_j}(y_i)}{p_{\mathrm{L}_k}(y_i)} &= \frac{(p_{\mathrm{T}}(y_i)_j + 0)/\|p_{\mathrm{T}}(y_i) + p_{\mathrm{GT}}(y_i)\|_1}{(p_{\mathrm{T}}(y_i)_k + 0)/\|p_{\mathrm{T}}(y_i) + p_{\mathrm{GT}}(y_i)\|_1} \\ &= \frac{p_{\mathrm{T}}(y_i)_j}{p_{\mathrm{T}}(y_i)_k}.\end{aligned} \tag{14}$$

If y_i is within the Top K, the ratio involving y_i also holds due to the uniform scaling by the L1 norm. Therefore, Constraint 2 is satisfied.

Constraint 3. For any $j \notin S$, $p_{\mathrm{T}}(y_i)_j = 0$ (due to Top K sparsification). Therefore, $p_{\mathrm{L}_j}(y_i) = 0$. For any $k \in S$, $p_{\mathrm{L}_k}(y_i)$ will be non-negative due to either a non-zero value in $p_{\mathrm{T}}(y_i)$ or the one-hot vector $p_{\mathrm{GT}}(y_i)$. Therefore, $p_{\mathrm{L}_j}(y_i) \leq p_{\mathrm{L}_k}(y_i)$ for all $j \notin S$ and $k \in S$, satisfying Constraint 3.

Constraint 4. By definition, the L1 norm normalization in Equation 7 ensures that the elements of $p_{\mathrm{L}}(y_i)$ sum to 1. Furthermore, since both $p_{\mathrm{T}}(y_i)$ and $p_{\mathrm{GT}}(y_i)$ have non-negative elements, $p_{\mathrm{L}}(y_i)$ will also have non-negative elements. The normalization then ensures that all elements are within the range [0, 1]. Thus, Constraint 4 is satisfied.

B Dataset Details

Table 7 provides a comprehensive overview of the datasets used in our study, detailing their sampled sizes, data sources, and associated references. The datasets include Socratic and StackExchange from (Yue et al., 2024), Camel-AI (covering math, physics, biology, and chemistry) from (Li et al., 2023), MathInstruct from (Jiang et al., 2024), GSM8K from (Cobbe et al., 2021), MetaMath and MetaMath-GSM8K from (Yu et al., 2023), and OpenMathInstruct2 from (Toshniwal et al., 2024). For OpenMathInstruct2, which contains 1M samples, we sampled 10K for evaluation.

C Logits-Based Dataset Example

Table 6 presents a logits-based label visualization for the sentence: "There are 12 inches in 1 foot, so Vlad's height is $6 \times 12 + 3 = 75$ inches. His sister's height is $2 \times 12 + 10 = 34$ inches."

There	\n\n			
are	is	’s		
<space>	two			
12	6	3	2	
inches				
in	per			
<space>	a	each	one	every
1				
foot				
,	.	.\n	.\n\n	so
so	and	therefore	<space>	
Vlad	we	<space>	multiply	convert
’s	is	’s		
height	<space>	total		
is	in	can	of	,
<space>	\\(	\n	$	(
6	12			
*	feet	(	x	\\
<space>	12			
12				
+	=			
<space>				
3				
=	inches			
<space>				
75	72	63		
inches	in			
.\n	.	and	,	.\n\n
His	V	Similarly	The	S
sister	younger	sisters		
’s	is			
height				
is				
<space>				
2				
*	*			
<space>				
12				
+				
<space>				
10				
=				
<space>				
34				
inches				
.\n	.	,	.\n\n	

Table 6: Example of the logits-based label of *There are 12 inches in 1 foot, so Vlad’s height is 6 * 12 + 3 = 75 inches. His sister’s height is 2 * 12 + 10 = 34 inches.*.

Dataset	Sampled Size	Data Source	Paper Source
Socratic	511k	TIGER-Lab/WebInstructSub	Yue et al. (2024)
StackExchange	291k	TIGER-Lab/WebInstructSub	Yue et al. (2024)
ScienceQA	100k	ibivibiv/science_qa	-
Camel-AI	50k	camel-ai/math	Li et al. (2023)
	20k	camel-ai/physics	Li et al. (2023)
	20k	camel-ai/biology	Li et al. (2023)
	20k	camel-ai/chemistry	Li et al. (2023)
MathInstruct	262k	TIGER-Lab/MathInstruct	Jiang et al. (2024)
GSM8K	7.5k	openai/gsm8k	Cobbe et al. (2021)
MetaMath	395k	meta-math/MetaMathQA	Yu et al. (2023)
MetaMath-GSM8K	240k	meta-math/MetaMathQA	Yu et al. (2023)
OpenMathInstruct2	10k	nvidia/OpenMathInstruct-2	Toshniwal et al. (2024)

Table 7: Size and Source of the datasets, including Socratic (Yue et al., 2024), StackExchange (Yue et al., 2024), Camel-AI (Li et al., 2023), MathInstruct (Jiang et al., 2024), GSM8K (Cobbe et al., 2021), MetaMath (Yu et al., 2023), MetaMath-GSM8K (Yu et al., 2023), and OpenMathInstruct2 (Toshniwal et al., 2024).