Explainable Token-level Noise Filtering for LLM Fine-tuning Datasets

浙江大学 ZHEJIANG UNIVERSITY

CONTENT

Introduction

- **We propose an explainable token-level noise filtering framework to filter noisy tokens during fine-tuning.**

- Large language models (LLMs) have achieved remarkable progress and delivered state-of-the-art performance across a wide range of applications.As a crucial step for adapting LLMs to specific downstream tasks, fine-tuning typically requires further training on corresponding datasets.However, a fundamental inconsistency exists between current fine-tuning datasets and the token-level optimization objectives of LLMs:most datasets are constructed at the sentence level, which introduces token-level noise and adversely affects final performance.

We propose XTF, an interpretable token-level noise filtering framework.XTF systematically scores each label token based on three proposed properties,and then masks the gradients of selected noisy tokens during fine-tuning,thereby effectively optimizing the performance of fine-tuned LLMs.

Background

❑ **Technical scenario: Existing sentence-level datasets are not aligned with the token prediction loss of LLMs.**

- The fundamental framework of LLMs is token-prediction, while existing fine-tuning datasets are designed at the sentence level, which is misaligned with the actual model inference and training processes.

Input: example in gsm8k

Weng earns $12 an hour for babysitting. Yesterday, she just did 50 minutes of babysitting. How much did she earn?

↓ sentence 2 sentence data

Output:

Weng earns 12/60 = $<<12/60=0.2>>0.2 per minute.
Working 50 minutes, she earned 0.2 x 50 = $<<0.2*50=10>>10.
10

There is more noise in non-professional datasets (e.g., in the medical and education domains).

{Input}	→ Weng
{Input} Weng	→ earns
{Input} Weng earns	→ 12
{Input} Weng earns 12	→ /
{Input} Weng earns 12/	→ 60
{Input} Weng earns 12/60	→ =
{Input} Weng earns 12/60 =	→ $
... ...	token-level prediction ↑

- **Optimization objective drift:** The introduction of noisy tokens deviates from the original optimization target, resulting in degraded performance.
- **Latent parameter deviation:** Noisy tokens cause disordered convergence directions and potentially induce task-irrelevant performance drops.

Related Work

❑ Technical Background: Lack of token-level data research for fine-tuning tasks

Data optimization methods：data augmentation and data filtering. These methods only optimize data at the sentence or sample level, resulting in coarse optimization granularity that cannot match the token-level training process；

Token interpretability methods focus on explaining the relationship between input tokens and output results during inference.However, such methods cannot account for the correlation between output labels and fine-tuning effectiveness.

Token-level training optimization: Existing works can be divided into three directions:：

- Knowledge distillation: Token-level knowledge distillation is more suitable for simple scenarios, while sentence-level distillation is appropriate for complex scenarios.（IJCAI’24）
- Human Preference Optimization (DPO): Compares token-level differences between the reference and the original output, and is only applicable to DPO；（ICML’24）
- **Translation Model: Observes token loss variations during training and adjusts attention dropout.：**
  - Translation model: Monitor the variation of token loss during training and adjust attention dropout accordingly.（ACL’23）
  - SLM: First train a reference model on high-quality corpora, then select important tokens based on the variation of loss values on tokens from normal training data.；（NIPS’24）

——There exists an implicit assumption: that the training corpus contains no noise, or that a noise-free high-quality dataset exists.

This assumption is objectively invalid. Constructing such a "high-quality dataset" for domain-specific fine-tuning requires extensive expert effort and is impractical.

Meanwhile, no research has proven that existing high-quality training sets (e.g., GSM8K, HumanEval) are entirely noise-free.

Technical Route

Technical Objectives

Theoretical Foundation and Noise Definition：**First, since there is a lack of research on the relationship between token-level data and fine-tuning performance, it is necessary to theoretically define the data requirements for fine-tuning tasks, and then define what constitutes a noisy token for such tasks.**

Parametric Evaluation and Filtering：**We explain the role of each token for the fine-tuning task through a parametric approach. After fixing the model and dataset, we evaluate the value of each token in the output labels within the dataset.**

Fine-tuning Performance Enhancement：**The essence of data optimization is to optimize the convergence direction during training.Thus, it can improve the model performance on the fine-tuning target tasks.Meanwhile, the changes in the model’s out-of-domain performance can be evaluated to clarify the potential adverse effects of fine-tuning noise.**

Technical Route

Technical Implementation

Theoretical Foundation and Noise Definition：According to the requirements of fine-tuning: learning new knowledge related to downstream tasks in the dataset based on the reasoning ability of the base model.We define three crucial properties for fine-tuning performance：

- **Reasoning Importance:** Evaluate whether the information is critical to forming contextual logic for the base model. Such information is valuable for preserving the reasoning ability of the base model.
- **Knowledge Shift:** Evaluate whether the information contains new knowledge for the base model. Existing common-sense knowledge does not need to be learned repeatedly.
- **Task Relevance:** Evaluate whether the information is relevant to the task objective.Low-relevance information contributes little to the fine-tuning task.

Intuitively insight：Only data that satisfies all three properties above is truly beneficial for fine-tuning tasks.Thus, the representation of fine-tuning noise data should be：

$$D_{\text{noise}} = (D_{RI\downarrow}) \cup (D_{KN\downarrow}) \cup (D_{TR}\downarrow)$$

Technical Route

❑ Technical Implementation

Theoretical Foundation and Noise Definition：

1. **Reasoning Importance** (RI)**:** Let $S_{RI}(t_i)$ be the reasoning importance score of token t_i, derived from an attention-based scoring function $\mathcal{A}_{\text{RI}}(\theta_0, x, y, i)$ which quantifies the token's structural role within the sequence $x+y$ as interpreted by $f_{\boldsymbol{\theta}_0}$. We say $t_i \in D_{RI\downarrow}$ (low reasoning importance) if $S_{RI}(t_i) \leq \tau_{RI}$.
   Idealized Interpretation: A token $t_i \in D_{RI\downarrow}$ does not significantly contribute to the core inferential chain or compositional semantic structure necessary for $f_{\boldsymbol{\theta}_{\text{opt}}}$ to map $x \to y^*$. Its contribution to the representation of y that determines task success is negligible. For example, if $y = h(y_{\text{core}}, y_{\text{superficial}})$ where y_{core} are essential components, $t_i \in y_{\text{superficial}}$ would have low RI.
2. **Knowledge Novelty** (KN)**:** Let $S_{KN}(t_i) = 1 - P_{\boldsymbol{\theta}_0}(t_i|x, t_{<i})$ be the knowledge novelty score. We say $t_i \in D_{KN\downarrow}$ (low knowledge novelty) if $S_{KN}(t_i) \leq \tau_{KN}$, which implies $P_{\boldsymbol{\theta}_0}(t_i|x, t_{<i}) \geq 1 - \tau_{KN}$.
   Idealized Interpretation: A token $t_i \in D_{KN\downarrow}$ has high conditional probability given the context under the base model $f_{\boldsymbol{\theta}_0}$. This implies that the information carried by t_i, $I(t_i; \theta_0|x, t_{<i}) = -\log_2 P_{\boldsymbol{\theta}_0}(t_i|x, t_{<i})$, is low. The token t_i represents information already well-assimilated by θ_0 and does not introduce substantial new information (reducing uncertainty) pertinent to the downstream task that $f_{\boldsymbol{\theta}_0}$ has not already captured.
3. **Task Relevance** (TR)**:** Let $S_{TR}(t_i) = 1 - \text{Normalize}(\mathcal{D}_{\text{dist}}(\mathcal{E}(t_i; \theta_0), \mathcal{V}(\text{Domain})))$ be the task relevance score, where $\mathcal{E}(t_i; \theta_0)$ is the embedding of t_i from $f_{\boldsymbol{\theta}_0}$, $\mathcal{V}(\text{Domain})$ is a representation of the task's semantic domain (e.g., centroid of domain-specific term embeddings), and $\mathcal{D}_{\text{dist}}$ is a distance metric in the embedding space. We say $t_i \in D_{TR\downarrow}$ (low task relevance) if $S_{TR}(t_i) \leq \tau_{TR}$.
   Idealized Interpretation: A token $t_i \in D_{TR\downarrow}$ lies in a region of the semantic embedding space that is distant from the manifold $\mathcal{M}_{\text{domain}}$ representing the core concepts of the target task. An ideal task-specialized model $f_{\boldsymbol{\theta}_{\text{opt}}}$ would primarily assign probability mass to tokens whose embeddings $\mathcal{E}(t; \theta_{\text{opt}})$ are within or near $\mathcal{M}_{\text{domain}}$ when generating task-aligned outputs y^*.

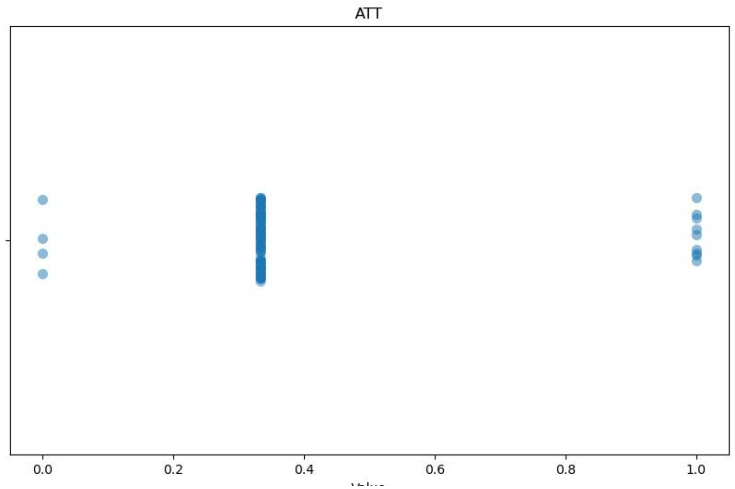
Reasoning importance score: Using the attention value of the token.

Knowledge Novelty Score: Using 1 minus the predicted probability of the token.

Task Relevance Score: 1 minus the average distance between the token and domain_words in the embedding space.

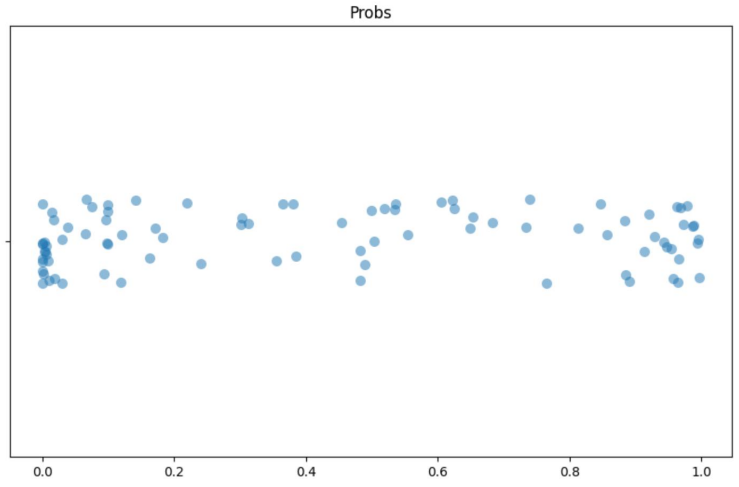
Technical Route

Token Score Distribution



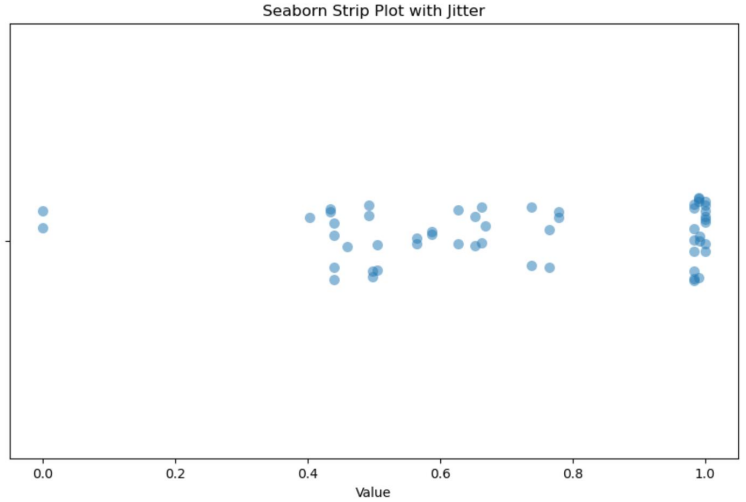
Filter extreme values to the left using the IQR method.

Attention scores often exhibit an extreme distribution: most tokens have identical attention scores, while a small fraction shows either very high or very low values.



Filter out tokens with a prediction probability greater than 99%.

The distribution of probs is relatively uniform and is mainly determined by dataset characteristics.For example, probs are generally low in text tasks, while they are relatively high in code tasks.



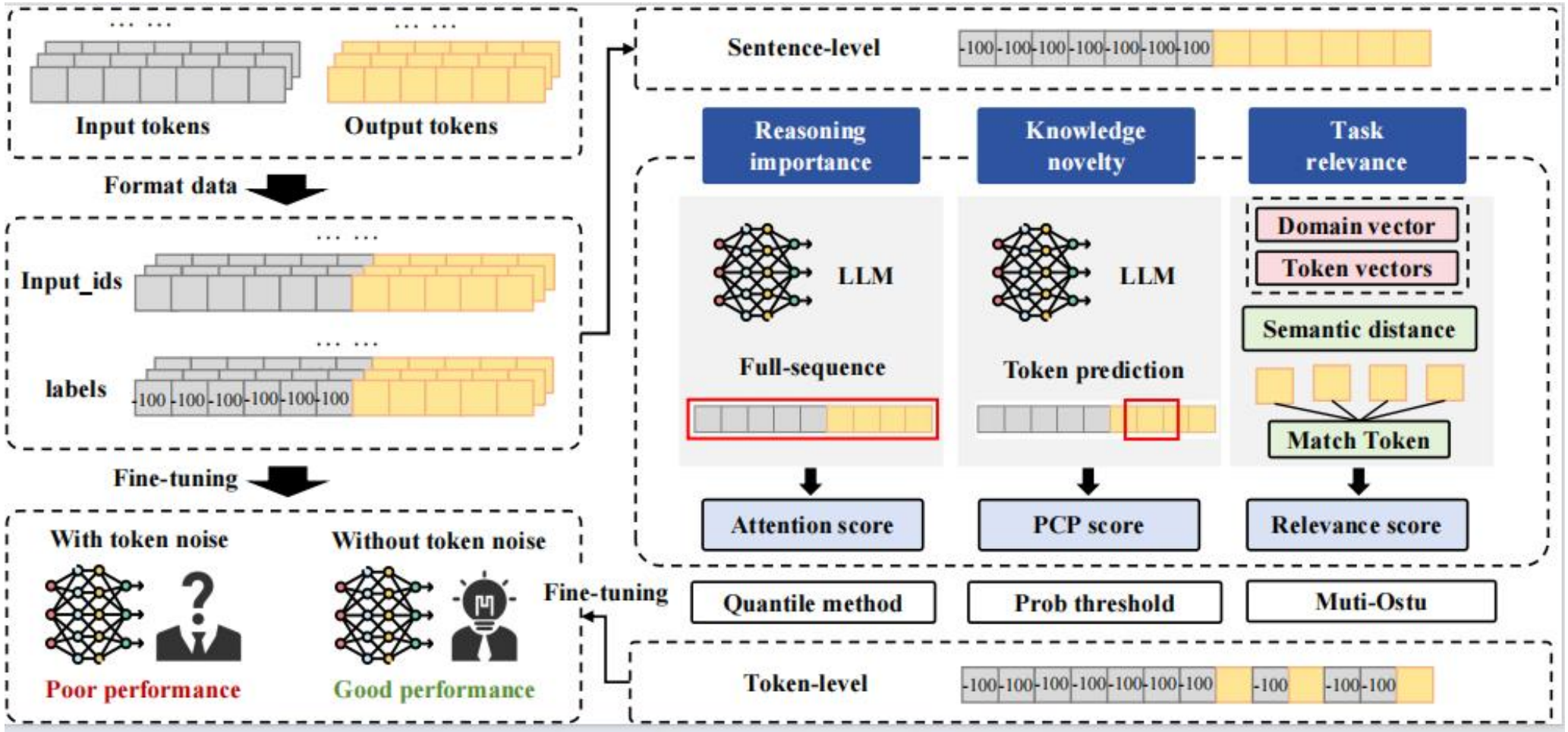
Multi-Otsu clustering for multi-class value partitioning

The distribution of relevance scores exhibits clustering characteristics.The minimum cluster consists of whitespace tokens, while the rest follow a hierarchical distribution based on semantic similarity, which can be filtered using clustering

Technical Route

Technical Implementation

Parameterized Representation and Noise Filtering:



Technical Route

ZHEJIANG UNIVERSITY

❑ Example of Noisy Token

Question: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?

Attention Importance

Analysis: Natalia sold 48/2 = <<48/2=24>>24 clips in May. Natalia sold 48+24 = <<48+24=72>> 72 clips altogether in April and May. Answer: 72.

Knowledge Novelty

Analysis: Natalia sold 48/2 = <<48/2=24>>24 clips in May. Natalia sold 48+24 = <<48+24=72>> 72 clips altogether in April and May. Answer: 72.

Task Relevance

Analysis: Natalia sold 48/2 = <<48/2=24>>24 clips in May. Natalia sold 48+24 = <<48+24=72>> 72 clips altogether in April and May. Answer: 72.

Together

Analysis: Natalia sold 48/2 = <<48/2=24>>24 clips in May. Natalia sold 48+24 = <<48+24=72>> 72 clips altogether in April and May. Answer: 72.

Experimental Results

❑ Experimental Setup

Datasets

We select three representative downstream tasks to evaluate the fine-tuning performance, including two mainstream tasks: mathematics and coding (which are widely used to evaluate large language models), as well as an important professional task: medicine. For the mathematics task, we use GSM8K for fine-tuning and evaluation. For the coding task, we perform fine-tuning on CodeExercise and evaluate using HumanEval. For the medical task, we employ PubMedQA for fine-tuning and evaluation.

LLMs

We select seven base large language models of different sizes from three prominent model families: DeepSeek, Llama, and Mistral. Specifically, for the DeepSeek family, we choose DeepSeek-R1-distilled-Qwen-1.5B, 7B, and 14B; for the Llama family, we select Llama-3.2-1B, 3B, and Llama-3.1-8B; and for the Mistral family, we adopt Mistral-v0.1-7B. All the selected large language models are raw base models without any fine-tuning.

baseline

We consider three conventional large language model fine-tuning methods and three data augmentation approaches to demonstrate the effectiveness of our XTF. Specifically, for the standard LLM fine-tuning implementations, we adopt Clean Accuracy (CA, using the original base model), normal fine-tuning (Normal), and fine-tuning with doubled training iterations (×2 epochs). The data augmentation methods filter data noise from different perspectives: Data Filtering (DF) removes noisy data at the sample level; Data Augmentation (DA) improves model robustness against noise by enriching the training data; and Selective Language Model training (SLM) performs token-level selective training based on changes in loss values.

Experimental Results

浙江大学 ZHEJIANG UNIVERSITY

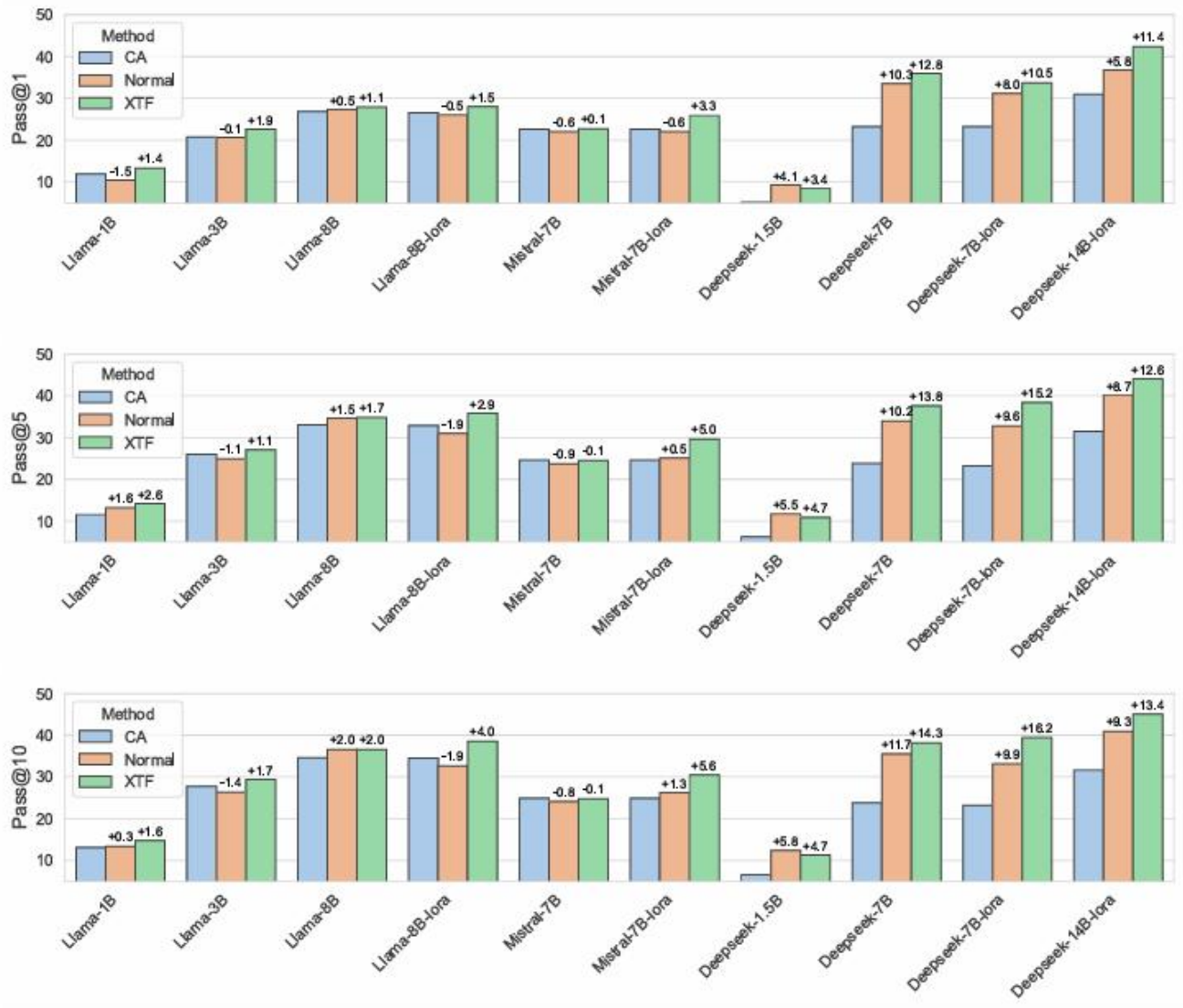
❑ Experimental Results on Mathematical and Medical Datasets

MATH: Fine-tuning and evaluate models on GSM8K									
Model	$\|\theta\|$	**LoRA**	**CA**	**Normal**	**×2 Ep**	**DF**	**DA**	**SLM**	**XTF**
Llama-3.2	1B	×	2.8	4.3	6.8	7.6	2.4	5.9	**8.7**
Llama-3.2	3B	×	3.9	25.8	33.4	36.9	27.1	38.8	**40.5**
Llama-3.1	8B	×	4.6	54.0	55.4	52.7	55.4	**60.3**	58.7
Llama-3.1	8B	✓	4.6	33.7	32.7	37.0	33.7	**37.9**	37.1
Mistral	7B	×	8.0	15.0	16.1	21.3	18.4	22.6	**29.1**
Mistral	7B	✓	8.0	23.4	20.7	25.6	21.8	21.3	**25.6**
Deepseek-distilled-qwen	1.5B	×	17.6	42.9	45.5	47.0	37.4	37.3	**56.2**
Deepseek-distilled-qwen	7B	×	37.9	63.0	68.1	65.5	56.5	63.8	**69.3**
Deepseek-distilled-qwen	7B	✓	37.9	61.3	60.4	67.9	62.0	68.2	**71.8**
Deepseek-distilled-qwen	14B	✓	34.5	47.6	50.3	52.4	50.5	49.3	**60.3**
Average	–	–	16.0	37.1	39.0	41.4	36.5	40.5	**45.7**

MEDICINE: Fine-tuning and evaluate models on PubMedQA									
Model	$\|\theta\|$	**LoRA**	**CA**	**Normal**	**×2 Ep**	**DF**	**DA**	**SLM**	**XTF**
Llama-3.2	1B	×	2.9	15.5	15.6	15.4	18.3	13.1	**18.5**
Llama-3.2	3B	×	6.6	35.5	33.7	29.8	**37.9**	36.1	36.5
Llama-3.1	8B	×	4.9	13.6	14.0	18.4	14.7	**22.9**	21.3
Llama-3.1	8B	✓	4.9	24.2	22.3	34.3	31.4	26.3	**37.9**
Mistral	7B	×	8.6	20.0	26.2	18.7	23.4	26.5	**32.0**
Mistral	7B	✓	8.6	15.5	18.4	15.6	13.1	17.5	**21.3**
Deepseek-distilled-qwen	1.5B	×	39.8	44.6	45.4	41.2	49.7	48.3	**50.8**
Deepseek-distilled-qwen	7B	×	42.7	50.5	42.1	55.4	52.7	51.3	**55.6**
Deepseek-distilled-qwen	7B	✓	**42.7**	35.9	37.8	33.6	38.4	39.0	41.7
Deepseek-distilled-qwen	14B	✓	44.6	48.5	43.4	51.3	47.0	53.9	**55.7**
Average	–	–	20.6	30.4	29.9	31.4	32.7	33.5	**37.1**

Experimental Results

❑ Experimental Results on the Code Dataset



Conclusion

- In this paper, we investigate the impact of training data on fine-tuning performance at the token level. We explore solutions to filter token-level noise from three dimensions: reasoning importance, knowledge novelty, and task relevance, so as to optimize the fine-tuning dataset, and propose the XTF method based on these insights.
- We conduct extensive experiments on three distinct downstream tasks and seven representative large language models.
- The results show that XTF achieves accuracy improvements of up to 13.3%, 13.7%, and 6.3% on the mathematical, medical, and code tasks respectively, and outperforms all baseline methods in overall performance, demonstrating its effectiveness in noise filtering and fine-tuning enhancement.