

# ROBUSTFT: Robust Supervised Fine-tuning for Large Language Models under Noisy Response

Junyu Luo<sup>♡</sup>, Xiao Luo<sup>♣</sup>, Kaize Ding<sup>◊</sup>, Jingyang Yuan<sup>♡</sup>, Zhiping Xiao<sup>♣</sup>, Ming Zhang<sup>♡</sup>

<sup>♡</sup> Peking University    <sup>♣</sup> University of California, Los Angeles

<sup>◊</sup> Northwestern University    <sup>♣</sup> University of Washington

luojunyu@stu.pku.edu.cn, xiaoluo@cs.ucla.edu, kaize.ding@northwestern.edu

patxiao@uw.edu, {yuanjy, mzhang\_cs}@pku.edu.cn

## Abstract

Supervised fine-tuning (SFT) plays a crucial role in adapting large language models (LLMs) to specific domains or tasks. However, as demonstrated by empirical experiments, the collected data inevitably contains noise in practical applications, which poses significant challenges to model performance on downstream tasks. Therefore, there is an urgent need for a noise-robust SFT framework to enhance model capabilities in downstream tasks. To address this challenge, we introduce a robust SFT framework (ROBUSTFT) that performs noise detection and relabeling on downstream task data. For noise identification, our approach employs a multi-expert collaborative system with inference-enhanced models to achieve superior noise detection. In the denoising phase, we utilize a context-enhanced strategy, which incorporates the most relevant and confident knowledge followed by careful assessment to generate reliable annotations. Additionally, we introduce an effective data selection mechanism based on response entropy, ensuring only high-quality samples are retained for fine-tuning. Extensive experiments conducted on multiple LLMs across five datasets demonstrate ROBUSTFT’s exceptional performance in noisy scenarios. Our code and data are publicly available.<sup>1</sup>

## 1 Introduction

Supervised fine-tuning (SFT) has emerged as a critical technique for optimizing Large Language Models’ (LLMs) capabilities, particularly in domain-specific tasks and adapting them to specific scenarios (Minaee et al., 2024; Raffel et al., 2020). High-quality scenario-specific data plays a vital role in enhancing model performance on downstream tasks (Xia et al., 2024; Jeong et al., 2024). While such data can be acquired through various means including human annotation (Li et al.,

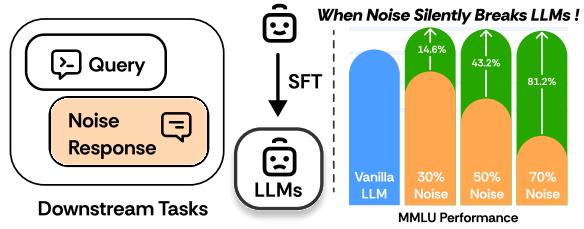


Figure 1: Impact of noisy data on LLM performance during SFT. Increasing noise levels deteriorates model performance, highlighting the critical need for noise-robust fine-tuning approaches.

2024), scenario-specific collection (Clark et al., 2019), and model-based self-labeling (Wang et al., 2024), these data sources inherently contain noise stemming from both human annotation errors and model hallucinations (Farquhar et al., 2024).

Data noise in scenario can have catastrophic effects on model performance. As shown in Figure 1, the MMLU (Hendrycks et al., 2020) evaluation results clearly demonstrate this degradation: as the proportion of noisy data increases, model accuracy shows a sharp decline. Specifically, with just 30% noise in the training data, the model’s performance deteriorates by 8.9% compared to the vanilla LLM baseline. This performance degradation becomes increasingly severe as noise levels rise further. These findings underscore the critical importance and practical value of developing noise-robust fine-tuning frameworks for LLMs to maintain reliable downstream performance. This motivates our central research question:

*Can LLMs detect inevitable noise and enhance data quality, to improve its performance on target tasks?*

The development of a noise-robust LLM fine-tuning framework encounters two major challenges. *First*, direct noise detection through LLM predictions proves unreliable due to model hallucinations and overconfidence, as validated by our empirical studies in Section 4. *Second*, while existing noise-

<sup>1</sup><https://github.com/luo-junyu/RobustFT>

robust methods work well for classification tasks with discrete label spaces (Yuan et al., 2024; Wang et al., 2023a), they are inadequate for LLM fine-tuning scenarios that require contextual and open-ended text generation. Traditional relabeling strategies not only fail to utilize valuable information in noisy generated responses. These challenges highlight the complexity of developing a framework that effectively leverages both model capabilities and data characteristics for robust noise detection and denoising in LLM fine-tuning.

In this paper, we propose ROBUSTFT (Noise-robust LLM Supervised Fine-Tuning), a framework for effective adaptation in downstream scenarios with noisy data. At its core, ROBUSTFT introduces multi-view noise detection and denoising strategies. For noise detection, ROBUSTFT employs a collaborative multi-expert system, incorporating *reasoning-enhanced* models to identify potentially noisy data effectively. For the identified noisy data, ROBUSTFT designs a denoising and data selection process. First, ROBUSTFT utilizes high-confidence data as contextual references for reliable relabeling of noisy samples. Subsequently, for both *context-enhanced* and *reasoning-enhanced* inference, ROBUSTFT employs Review Agent to examine and synthesize responses. Finally, by computing confidence scores based on model response entropy and excluding low-confidence samples, we obtain a denoised fine-tuning dataset that facilitates model adaptation to downstream tasks. Overall, by combining noise detection and denoising processes, ROBUSTFT effectively enhances the quality of the fine-tuning dataset while maximizing data utility. We validate ROBUSTFT’s effectiveness through extensive experiments across five datasets, spanning both general and domain-specific tasks with varying noise levels. Through comprehensive comparative analyses and ablation studies, we demonstrate the superiority of our approach.

Our contributions can be summarized as follows:

- **New Perspective.** We investigate the critical yet understudied challenge of noise-robust supervised fine-tuning for LLMs, which aligns more closely with real-world scenarios where noise is inevitable.
- **Principled Methodology.** We design a self-contained framework to leverage the intrinsic interactions between models and data for effective noise detection and denoising, eliminating dependencies on external models or resources.

- **Superior Performance.** ROBUSTFT exhibits robust performance across diverse noise conditions, demonstrating significant improvements on three open-source LLMs across both general and domain-specific tasks, which validates its broad applicability and practical value.

## 2 Preliminary

### 2.1 Real-world Challenge

In practical applications of Large Language Models (LLMs), our objective extends beyond enhancing their general capabilities to improving their performance on downstream tasks. To achieve this, we utilize Supervised Fine-Tuning (SFT) to optimize an LLM  $\mathcal{M}$  for a target downstream task  $\mathcal{D}_{task} = \{q_i, y_i\}_{i=1}^N$ , where  $q_i$  denotes the query and  $y_i$  is the expected response. The model’s performance is enhanced by minimizing the loss between its predictions and the expected outputs.

However, the effectiveness of SFT is heavily dependent on the quality of the downstream task data (Bhatt et al., 2024; Xia et al., 2024). Various factors, including annotation errors, data processing inconsistencies, and model hallucinations, can introduce both random and systematic noise into downstream datasets  $\mathcal{D}_{task}$ . Our empirical studies in Section 4 demonstrate that 30% noise in the training data can lead to an 8.9% degradation on downstream tasks. Therefore, developing robust mechanisms for noise detection and mitigation during the SFT process, particularly ones that can effectively handle open-ended text generation, is crucial and holds significant practical value for optimizing LLM performance.

### 2.2 Problem Definition

As discussed above, during the fine-tuning of LLMs on downstream tasks, the training data contains both correctly and incorrectly labeled data pairs. Our primary objective is to develop an effective mechanism for identifying these mislabeled instances. Furthermore, we aim to leverage both the model’s capabilities and contextual information within the dataset to denoise incorrectly labeled data pairs where possible. Through this process, we seek to construct a refined dataset with reduced noise levels. Ultimately, this curated dataset enables more effective enhancement of LLM performance on downstream tasks.

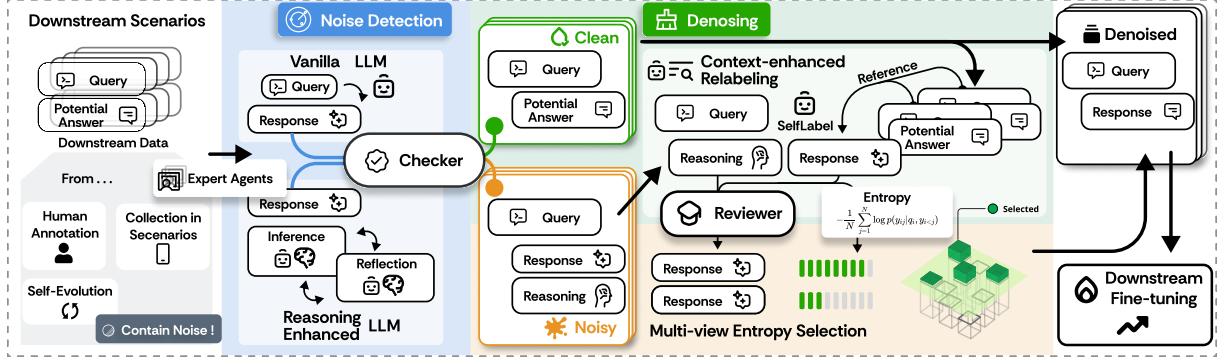


Figure 2: Overview of ROBUSTFT. Our ROBUSTFT enhances model performance through a two-stage noise *detection-and-denoising* framework, leveraging collaborative learning among expert LLMs for noise detection and context-enhanced reasoning for data denoising, ultimately enabling robust downstream fine-tuning.

### 3 Methodology

#### 3.1 Overview

Adapting and fine-tuning Large Language Models (LLMs) in real-world scenarios presents significant challenges, particularly due to the presence of noise in downstream task datasets that can compromise model performance. Our approach addresses this challenge through a systematic framework comprising noise *detection-and-denoising* mechanisms to prevent performance degradation.

For *noise detection*, we leverage the consensus among multiple expert LLMs and employ a Checker to identify noisy samples. For *denoising*, we employ a two-pronged approach: first, we utilize context-enhanced reasoning with clean samples to relabel noisy instances through a Review Agent; second, we implement a perplexity-based data selection mechanism to exclude samples with low confidence scores. As demonstrated in Figure 2, this dual-process framework effectively mitigates noise-induced performance deterioration.

#### 3.2 Noise Detection

Effective noise identification is crucial for handling noisy data in downstream tasks. In our approach, we leverage collaborative learning among multiple LLMs to uncover potentially noisy samples, enabling a more robust detection mechanism.

Initially, we utilize the base LLM to generate predictions for all data samples:

$$\hat{y}_i = \mathcal{M}(q_i), \quad (1)$$

where  $q_i$  represents the query,  $\mathcal{M}$  denotes the LLM and  $\hat{y}_i$  is the base prediction.

For internal noise detection, we introduce a reasoning-enhanced LLM that iteratively combines

reasoning and reflection processes. This LLM first performs step-by-step reasoning, followed by self-reflection on its reasoning path, and iterates between these two stages to achieve superior reasoning capabilities. For each data sample, this iterative process can be formalized as:

$$\hat{y}_i^{\text{reas}} = \mathcal{M}_{\text{Reas}}(q_i, \mathcal{M}_{\text{Refl}}(\mathcal{M}_{\text{Reas}}(q_i, \dots))), \quad (2)$$

where  $\hat{y}_i^{\text{reas}}$  represents the final prediction,  $\mathcal{M}_{\text{Reas}}$  and  $\mathcal{M}_{\text{Refl}}$  denote the reasoning and reflection LLMs, respectively, with each reflection stage evaluating and refining the previous reasoning output.

To ensure prediction reliability, we implement a consistency-based Checker mechanism that analyzes multiple prediction sources: the original label ( $y_i$ ), the base LLM prediction ( $\hat{y}_i$ ), and the reasoning-enhanced prediction ( $\hat{y}_i^{\text{reas}}$ ). This mechanism evaluates the agreement among these predictions through a consistency metric:

$$r_i = \text{Checker}(y_i, \hat{y}_i, \hat{y}_i^{\text{reas}}) \in \{0, 1\}, \quad (3)$$

where  $r_i = 1$  indicates high prediction consistency (reliable sample) and  $r_i = 0$  indicates prediction inconsistency (potentially noisy sample). Based on this consistency evaluation, we partition the dataset into clean samples  $\mathcal{D}_{\text{clean}} = \{(q_i, y_i) | r_i = 1\}$  and potentially noisy samples  $\mathcal{D}_{\text{noise}} = \{(q_i, y_i) | r_i = 0\}$  for subsequent denoising treatment.

#### 3.3 Data Denoising

For the potentially noisy dataset  $\mathcal{D}_{\text{noise}}$ , we employ a context learning approach for data relabeling, leveraging external knowledge to reduce noise in the data. Specifically, we project queries from both the reliable dataset  $\mathcal{D}_{\text{clean}}$  and the potentially noisy dataset  $\mathcal{D}_{\text{noise}}$  into a shared latent space:

$$h_i = \text{Encoder}(q_i) \in \mathbb{R}^d, \quad (4)$$

where  $h_i$  represents the  $d$ -dimensional latent representation of query  $q_i$  obtained through the encoder network.

During inference, for each noisy sample, we retrieve the  $k$  most similar samples from the reliable dataset as context for reasoning:

$$\hat{y}_i^{\text{cont}} = \mathcal{M} \left( q_i \mid \{(q_j, y_j)\}_{j \in \mathcal{N}_k(q_i, \mathcal{D}_{\text{clean}})} \right), \quad (5)$$

where  $\mathcal{N}_k(q_i, \mathcal{D}_{\text{clean}})$  denotes the indices of the  $k$  most similar samples to  $q_i$  in  $\mathcal{D}_{\text{clean}}$  based on their latent representations.

By incorporating external context, we enable the model to generate more reliable responses  $\hat{y}_i^{\text{cont}}$ . Combined with the previously obtained reasoning-enhanced predictions  $\hat{y}_i^{\text{reas}}$ , we introduce a Review Agent to evaluate and relabel the data:

$$\tilde{y}_i = \text{Review}(q_i, \hat{y}_i^{\text{cont}}, \hat{y}_i^{\text{reas}}). \quad (6)$$

Through the Review Agent’s assessment and synthesis, we obtain the relabeled predictions  $\tilde{y}_i$ , forming the denoised dataset  $\mathcal{D}_{\text{denoise}}$ . However, considering the potential for model errors and uncertainties, we must implement a data selection mechanism for the self-annotated denoised dataset to ensure quality and reliability.

### 3.4 Data Selection

While our denoising process generates a refined dataset  $\mathcal{D}_{\text{denoise}}$  through self-annotation, ensuring the quality of these auto-labeled samples remains crucial. To maintain high data quality and prevent error propagation during subsequent training, we introduce a confidence-based filtering mechanism leveraging entropy metrics. This approach enables us to quantitatively assess the uncertainty in context-enhanced predictions and retain only the most confident samples.

The entropy score for each context-enhanced response is computed as:

$$H(\hat{y}_i^{\text{cont}}) = -\frac{1}{N} \sum_{j=1}^N \log p(y_{ij}|q_i, y_{i < j}), \quad (7)$$

where  $p(y_{ij}|q_i, y_{i < j})$  represents the model’s prediction probability for the  $j$ -th token conditioned on the input query and previous tokens, and  $N$  denotes the sequence length. Lower entropy scores indicate higher model confidence and more deterministic predictions. Based on these scores, we rank and filter the samples to form our final selected dataset:

$$D_{\text{select}} = \{(q_i, \tilde{y}_i) \mid \text{rank}(H(\hat{y}_i^{\text{cont}})) \leq \beta | D_{\text{denoise}}\} \quad (8)$$

---

### Algorithm 1 Algorithm of ROBUSTFT

---

**Require:** Task dataset  $\mathcal{D}_{\text{task}}$ , LLM  $\mathcal{M}$ ;  
**Ensure:** Fine-tuned LLM  $\mathcal{M}'$ ;

- 1: //Noise Detection
- 2: Generate base predictions  $\hat{y}_i$  using  $\mathcal{M}$
- 3: Generate reasoning-enhanced predictions  $\hat{y}_i^{\text{reas}}$  via iterative reasoning-reflection
- 4: Use Checker to identify reliable samples
- 5: Split data into  $\mathcal{D}_{\text{clean}}$  and  $\mathcal{D}_{\text{noise}}$
- 6: //Data Denoising
- 7: **for** each sample in  $\mathcal{D}_{\text{noise}}$  **do**
- 8:     Generate context-enhanced prediction  $\hat{y}_i^{\text{cont}}$
- 9:     Use Review to generate denoised label  $\tilde{y}_i$
- 10: **end for**
- 11: //Data Selection
- 12: Calculate entropy scores for denoised samples
- 13: Select top- $\beta$  confident samples to form  $\mathcal{D}_{\text{select}}$
- 14: Fine-tune  $\mathcal{M}$  on  $\mathcal{D}_{\text{ft}}$  to obtain  $\mathcal{M}'$

---

where  $\beta$  controls the selection ratio, which defaults to 50% and will be validated in Section 4.3.2.

Through this process, we obtain  $\mathcal{D}_{\text{select}}$ , which demonstrates reduced noise levels and higher confidence scores through the combined application of denoising relabeling and selective filtering.

### 3.5 Summary

Through the integration of the processes described above, we combine the reliable dataset  $\mathcal{D}_{\text{clean}}$  and the selected denoised dataset  $\mathcal{D}_{\text{select}}$  to form our final fine-tuning dataset  $\mathcal{D}_{\text{ft}} = \mathcal{D}_{\text{clean}} \cup \mathcal{D}_{\text{select}}$ . Then, we fine-tune the LLM on  $\mathcal{D}_{\text{ft}}$ :

$$\mathcal{M}' = \arg \min_{\mathcal{M}} \mathbb{E}_{(q,y) \sim \mathcal{D}_{\text{ft}}} [-\log p_{\mathcal{M}}(y|q)], \quad (9)$$

where  $\mathcal{M}'$  represents the evolved model trained on the noise-reduced downstream task dataset. The complete algorithm is summarized in Algorithm 1.

## 4 Experiment

### 4.1 Experiment Setup

#### 4.1.1 Datasets

We conducted comprehensive evaluations on five diverse benchmark datasets: MMLU (Hendrycks et al., 2020), ARC (Clark et al., 2018), PubMedQA (Jin et al., 2019), Drop, and FPB (Malo et al., 2014). These datasets span multiple domains and task types: MMLU and ARC evaluate general knowledge across various academic disciplines;

Method	MMLU			ARC			PubMedQA			Drop			FPB		
	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%
Vanilla	65.3	65.3	65.3	82.7	82.7	82.7	72.0	72.0	72.0	87.2	87.2	87.2	75.5	75.5	75.5
Hermes-3	65.5	65.5	65.5	68.7	68.7	68.7	64.8	64.8	64.8	87.1	87.1	87.1	59.4	59.4	59.4
Tulu-3	55.7	55.7	55.7	73.3	73.3	73.3	63.3	63.3	63.3	85.3	85.3	85.3	54.5	54.5	54.5
SelfLabel	64.7	64.7	64.7	82.1	82.1	82.1	71.8	71.8	71.8	86.8	86.8	86.8	82.8	82.8	82.8
SFT	59.5	47.5	37.3	70.7	61.7	47.5	66.4	36.7	32.8	85.3	78.6	66.4	79.7	58.4	34.9
NoiseAL	66.3	65.5	66.1	84.0	83.6	83.4	74.2	72.2	71.8	86.8	84.3	82.1	81.1	78.5	72.8
SelfRAG	65.3	65.4	64.1	83.1	82.7	82.0	63.2	60.2	57.0	86.5	85.5	83.1	83.8	76.2	68.2
SelfSelect	59.1	53.4	44.0	76.8	72.1	62.6	57.8	46.0	22.6	86.2	78.8	64.4	79.8	58.4	32.0
<b>Ours</b>	<b>68.2</b>	<b>68.0</b>	<b>67.6</b>	<b>84.9</b>	<b>84.7</b>	<b>84.1</b>	<b>75.8</b>	<b>75.6</b>	<b>75.0</b>	<b>90.3</b>	<b>88.5</b>	<b>87.9</b>	<b>84.4</b>	<b>80.5</b>	<b>76.2</b>
↑ vs. Vanilla	4.4	4.1	3.5	2.7	2.4	1.7	5.3	5.0	4.2	3.6	1.5	0.8	11.8	6.6	0.9
↑ vs. SFT	14.6	43.2	81.2	20.1	37.3	77.1	14.2	106	129	5.9	12.6	32.4	5.9	37.8	110

Table 1: **Performance comparison** under different noise rates with Llama-3.1 8B. Best results are shown in **bold**. Numbers in the last two rows show relative improvements (%).

PubMedQA tests biomedical reasoning capabilities; Drop assesses numerical reasoning and reading comprehension; and FPB examines financial domain expertise. For each dataset, we constructed experiments with different noise rates (*i.e.*, 30%, 50%, and 70%) to evaluate model performance under different scenarios.

#### 4.1.2 Backbones and Baselines

**Base Models.** We employed diverse model architectures, including Gemma2-9B (Team et al., 2024) and Llama3.1-8B (Dubey et al., 2024), along with models of varying parameter sizes such as Llama3.2-3B (Dubey et al., 2024).

**Baselines.** To comprehensively validate our method’s effectiveness, we implemented several baseline approaches: (1) Vanilla: direct model inference; (2) SFT-enhanced solutions utilizing supplementary data to improve LLM performance, including Hermes-3 (Teknium et al., 2024) and Tulu-3 (Lambert et al., 2024)<sup>2</sup>; (3) Standard SFT (Hu et al., 2021) using potentially noisy training data; (4) Denoising approaches, including the state-of-the-art NoiseAL (Yuan et al., 2024) and LLM-based denoising methods such as SelfLabel and SelfSelect; (5) Self-enhancement methods like SelfRAG (Lewis et al., 2020), which augments inference context using training data. Detailed baseline implementations are provided in the Appendix.

#### 4.1.3 Implementation Details

We partitioned each dataset into training and test sets, introducing varying degrees of noise perturba-

tion in the training data. For model fine-tuning, we employed Low-Rank Adaptation (LoRA) (Hu et al., 2021) implemented through Llama-factory (Zheng et al., 2024) across all open-source models. The fine-tuning process was conducted for 2 epochs. We set the  $n = 4$  and  $\theta = 50\%$ , with further parameter analysis planned for subsequent experiments. The implementation code is available in our anonymous repository. Comprehensive data and training configurations are detailed in the Appendix.

### 4.2 Main Result

#### 4.2.1 Comparison with Baselines

Our comparative experiments with Llama3.1-8B revealed several significant findings. ROBUSTFT consistently demonstrated superior performance across all datasets. The experimental results yielded the following key insights.

**Noise management is critical in LLM fine-tuning.** The SFT results clearly demonstrate that direct fine-tuning with noisy data substantially degrades model performance, emphasizing the necessity for robust noise detection and removal.

**LLMs exhibit limited inherent noise detection capabilities.** SelfSelect’s inferior performance compared to SFT indicates that LLMs cannot effectively identify noise, necessitating specialized noise detection and removal mechanisms.

**Enhanced SFT approaches lack consistent improvement.** Methods like Tulu-3 and Hermes-3 failed to show uniform performance improvements across downstream tasks, suggesting the need for task-specific LLM adaptation strategies.

**Inference enhancement methods show modest gains.** Notably, these approaches achieved some

<sup>2</sup>Hermes-3: <https://huggingface.co/NousResearch/Hermes-3-Llama-3.1-8B>. Tulu-3: <https://huggingface.co/allenai/Llama-3-Tulu-3-8B-SFT>

Model	MMLU			ARC			PubMedQA			Drop			FPB		
	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%
<i>Llama3.2 3B</i>															
Vanilla	54.9	54.9	54.9	72.4	72.4	72.4	57.8	57.8	57.8	71.0	71.0	71.0	39.9	39.9	39.9
SFT	55.0	48.4	38.3	66.1	58.5	42.9	63.2	49.2	37.5	77.3	73.7	61.3	56.2	49.4	31.3
<b>Ours</b>	<b>58.5</b>	<b>58.2</b>	<b>57.9</b>	<b>74.6</b>	<b>74.3</b>	<b>72.6</b>	<b>68.9</b>	<b>67.9</b>	<b>67.9</b>	<b>78.9</b>	<b>77.6</b>	<b>75.6</b>	<b>66.1</b>	<b>59.4</b>	<b>46.8</b>
<i>Llama3.1 8B</i>															
Vanilla	65.3	65.3	65.3	82.7	82.7	82.7	72.0	72.0	72.0	87.2	87.2	87.2	75.5	75.5	75.5
SFT	59.5	47.5	37.3	70.7	61.7	47.5	66.4	36.7	32.8	85.3	78.6	66.4	79.7	58.4	34.9
<b>Ours</b>	<b>68.2</b>	<b>68.0</b>	<b>67.6</b>	<b>84.9</b>	<b>84.7</b>	<b>84.1</b>	<b>75.8</b>	<b>75.6</b>	<b>75.0</b>	<b>90.3</b>	<b>88.5</b>	<b>87.9</b>	<b>84.4</b>	<b>80.5</b>	<b>73.2</b>
<i>Gemma2 9B</i>															
Vanilla	70.3	70.3	70.3	90.2	90.2	90.2	66.4	66.4	66.4	90.7	90.7	90.7	83.1	83.1	83.1
SFT	63.6	52.1	40.3	77.9	64.6	55.0	61.7	39.8	30.4	88.8	80.5	67.3	88.1	60.7	35.6
<b>Ours</b>	<b>72.5</b>	<b>72.1</b>	<b>71.3</b>	<b>91.8</b>	<b>91.5</b>	<b>90.4</b>	<b>70.8</b>	<b>68.8</b>	<b>66.8</b>	<b>91.9</b>	<b>91.8</b>	<b>90.9</b>	<b>91.8</b>	<b>80.8</b>	<b>87.7</b>

Table 2: **Performance comparison** across different model architectures and noise rates. Best results for each model are shown in **bold**.

Variant	MMLU			ARC		
	30%	50%	70%	30%	50%	70%
Llama3.1-8B						
<b>ROBUSTFT</b>	68.2	68.0	67.6	84.9	84.7	84.1
w/o Selection	65.7	65.1	64.6	83.2	83.0	82.8
w/o Checker	65.3	65.0	64.9	82.7	82.6	82.2
w/o Reviewer	68.0	67.7	67.1	84.5	84.3	84.0
w/o CER	67.7	67.7	67.0	84.6	84.1	83.9
w/o REL	67.4	67.2	66.9	84.1	83.9	83.6

Table 3: **Ablation study** showing the impact of different noise rates (30%, 50%, 70%) on model variants across MMLU and ARC benchmarks.

performance improvements despite potential noise in context data, though the improvements were not comparable to our method’s results.

**Denoising approaches demonstrate mixed results.** While methods such as NoiseAL and SelfLabel show noise resistance and improvements on some datasets, they exhibit degradation on others.

#### 4.2.2 Comparison across Architectures and Parameter Sizes

We conducted extensive experiments across multiple model architectures (Llama3.2-3B, Llama3.1-8B, and Gemma2-9B), as shown in Table 2. Our investigation revealed several noteworthy insights:

##### **Larger models are not inherently more robust.**

Contrary to common intuition, increased parameter count does not correlate with better noise resistance. In fact, general-purpose large models may be more susceptible to noise during domain-specific fine-tuning due to their lack of domain priors.

##### **Transformation mechanism from general models**

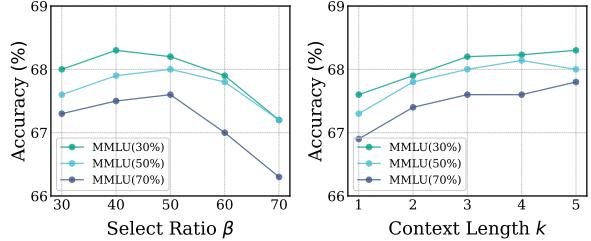


Figure 3: **Sensitivity analysis** on MMLU under different  $\beta$  and  $k$  with varying noise levels.

**to domain experts.** While Gemma2-9B showed strong general capabilities, it initially performed worse on domain-specific tasks. However, after ROBUSTFT, it effectively adapted to these domains and outperformed Llama3.1-8B, demonstrating the importance of denoising in LLM adaptation.

**Critical importance of denoising for smaller models.** Smaller models benefit more significantly from denoising strategies during domain-specific training. Our experiments show that effective denoising mechanisms can substantially mitigate the performance gaps of smaller models in downstream tasks.

#### 4.3 Analysis and Discussions

##### 4.3.1 Ablation Study

We conducted ablation experiments on RobustFT across different noise levels (30%, 50%, 70%) using MMLU and ARC datasets. The results reveal several key findings: **(1)** The complete RobustFT framework consistently achieves optimal performance across all settings, validating its effectiveness. **(2)** The Selection component proves crucial, as its removal leads to substantial per-

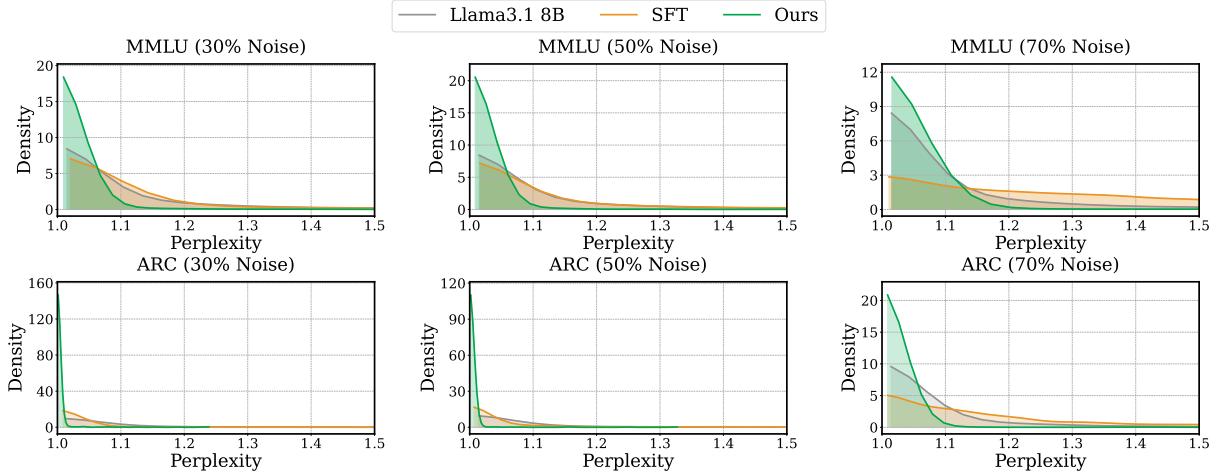


Figure 4: **Perplexity analysis** of ROBUSTFT on MMLU and ARC with varying noise levels.

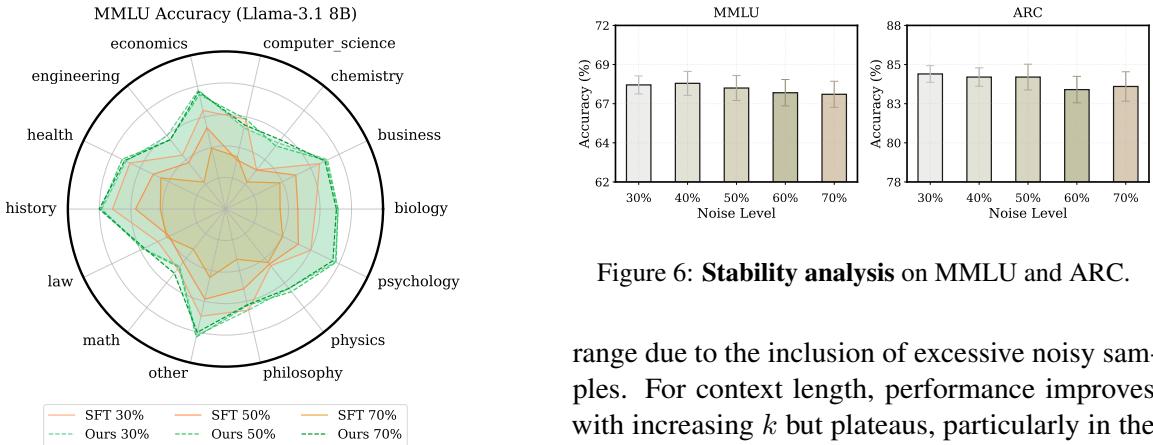


Figure 5: **Category-wise performance** of ROBUSTFT.

formance drops (*e.g.*, accuracy decreases from 68.2 to 65.7 on MMLU with 30% noise). **(3)** The Checker component significantly contributes to model performance, particularly on the ARC dataset, demonstrating the effectiveness of our multi-model collaborative noise detection. **(4)** While the Reviewer component shows modest impact, it still contributes to overall data quality. **(5)** Both Context-Enhanced Relabeing (CER) and Reasoning-Enhanced LLM (REL) components prove essential, with their removal leading to notable performance degradation, highlighting the importance of our multi-experts collaborative mechanisms in handling noisy data.

### 4.3.2 Sensitivity Analysis

We conducted sensitivity analysis of ROBUSTFT on MMLU under different noise levels. As shown in Figure 3, we examine the impact of selection ratio  $\beta$  and context length  $k$ . The results show that model performance peaking at  $\beta = 40 - 50\%$ , with performance degrading significantly beyond this

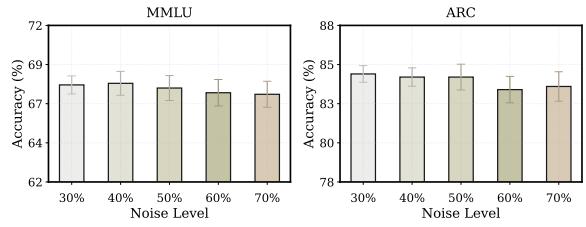


Figure 6: **Stability analysis** on MMLU and ARC.

range due to the inclusion of excessive noisy samples. For context length, performance improves with increasing  $k$  but plateaus, particularly in the range of  $k = 3 - 5$ , suggesting that moderate  $k$  provide sufficient reasoning support. These findings validate our default parameter choices ( $\beta = 50\%$ ,  $k = 3$ ) without requiring extensive hyperparameter search, as our primary focus was on demonstrating the framework’s overall effectiveness.

### 4.3.3 Perplexity Analysis

We conducted perplexity analysis of the models, as shown in Figure 4, revealing several key findings: (1) Noise significantly increases perplexity, as evidenced in both SFT and vanilla models. In contrast, ROBUSTFT maintains relatively low perplexity levels even with increased noise, demonstrating its robustness. (2) The vanilla model exhibits flatter and more dispersed perplexity distributions, indicating frequent uncertainty in predictions. ROBUSTFT effectively concentrates perplexity in lower ranges, suggesting more confident and reliable predictions. (3) The method shows consistency across datasets, with similar perplexity reduction patterns observed on both MMLU and ARC, validating its generalizability across different domains.

#### 4.3.4 Category-wise Performance Analysis

We analyzed performance across MMLU categories, as shown in Figure 5. (1) The impact of noise varies significantly across different knowledge domains, with knowledge-intensive categories such as History, Healthcare, and Law experiencing more severe performance degradation under noisy conditions. (2) ROBUSTFT demonstrates balanced and expanded performance across all categories, achieving comprehensive noise resistance rather than isolated improvements, as evidenced by its smooth and expanded radar plot.

#### 4.3.5 Stability Analysis

We evaluated the inference stability of models under different noise conditions, as shown in Figure 6. Specifically, we employed GPT-4o to rephrase the instructions and conducted five independent tests, reporting both mean performance and standard deviation. Results show that ROBUSTFT maintains consistent performance, with only minimal variance increase at higher noise rates.

## 5 Related Work

### 5.1 Noisy Label Learning

Noisy label learning has been a fundamental challenge in NLP (Yuan et al., 2024; Kim et al., 2024; Sun et al., 2023; Qi et al., 2023; Merdjanovska et al., 2024; Xu et al., 2024; Liang et al., 2024), primarily focusing on learning from text classification data containing label noise. Existing approaches can be categorized into three main strategies: (1) Sample selection methods (Qiao et al., 2022) that identify clean samples using fixed thresholds, (2) Label correction techniques (Sohn et al., 2020; Zhang et al., 2021) that rectify original labels based on model predictions, and (3) Consistency regularization approaches (Zhuang et al., 2023; Northcutt et al., 2021) that leverage prediction consistency under different perturbations for label refinement.

**Challenges in LLM Era.** These conventional methods are primarily designed for well-defined scenarios, with finite discrete label spaces, making them less effective for open-ended generation problems. Moreover, LLMs' tendency towards hallucination poses significant challenges in noise detection and correction. To address these limitations, ROBUSTFT introduces a novel framework specifically designed for noise-robust downstream fine-tuning of LLMs, moving beyond these constraints.

## 5.2 Toxicity Attacks and Defense

The vulnerability of LLMs to adversarial attacks through toxic and harmful data during post-training stages has garnered significant attention (Huang et al., 2024). Current defense mechanisms primarily focus on several key strategies: distance-based regularization (Mukhoti et al., 2023; Wei et al., 2024), alignment data mixing (Bianchi et al., 2023), prompt engineering (Lyu et al., 2024), and data filtering (Choi et al., 2024). Different from these methods, ROBUSTFT takes a different approach by emphasizing detection and relabeling mechanisms to prevent performance degradation caused by noisy data introduction, rather than specifically defending against toxic content.

## 5.3 Self-Evolution and LLM Data Selection

Recent advances in Large Language Models (LLMs) (Zhao et al., 2023) have emphasized the critical role of data quality in Supervised Fine-Tuning (SFT) (Taori et al., 2023; Longpre et al., 2023). Current research primarily explores two approaches: downstream data selection (Bhatt et al., 2024; Xia et al., 2024; Bukharin and Zhao, 2023) and data synthesis (Mukherjee et al., 2023; Chung et al., 2024) for improved instruction following. To reduce dependence on annotated data, researchers have developed self-evolution methods through self-instruction (Wang et al., 2023b) and self-play (Tu et al., 2024), enabling models to learn with minimal supervision. Additionally, SemiEvol (Luo et al., 2024) has demonstrated promising progress by combining a small amount of labeled data with large-scale unlabeled data to enhance LLM performance on downstream tasks. While existing work focuses on instruction selection (Parkar et al., 2024) and self-training mechanisms (Wang et al., 2024), ROBUSTFT takes a distinct approach by leveraging noisy real-world data for model self-training to enhance downstream performance.

## 6 Conclusion

In this work, we address the practical challenge of handling noisy data in downstream LLM applications, a critical issue that has been unexplored in previous research. We propose a novel noise detection and denoising framework ROBUSTFT, which is specifically designed for LLMs. Our approach leverages a multi-expert collaborative mechanism for noise detection, enhanced by a reasoning-enhanced process. Furthermore, we implement

context-enhanced reasoning for data relabeling and utilize response entropy for data selection. The effectiveness of ROBUSTFT is consistently demonstrated across various datasets and noise scenarios.

## References

- Gantavya Bhatt, Yifang Chen, Arnav M Das, Jifan Zhang, Sang T Truong, Stephen Mussmann, Yinglun Zhu, Jeffrey Bilmes, Simon S Du, Kevin Jamieson, et al. 2024. An experimental design framework for label-efficient supervised finetuning of large language models. *arXiv preprint arXiv:2401.06692*.
- Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. 2023. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. *arXiv preprint arXiv:2309.07875*.
- Alexander Bukharin and Tuo Zhao. 2023. Data diversity matters for robust instruction tuning. *arXiv preprint arXiv:2311.14736*.
- Hyeong Kyu Choi, Xuefeng Du, and Yixuan Li. 2024. Safety-aware fine-tuning of large language models. *arXiv preprint arXiv:2410.10014*.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.
- Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. 2024. Harmful fine-tuning attacks and defenses for large language models: A survey. *arXiv preprint arXiv:2409.18169*.
- Daniel P Jeong, Zachary C Lipton, and Pradeep Ravikumar. 2024. Llm-select: Feature selection with large language models. *arXiv preprint arXiv:2407.02694*.
- Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577.
- Yeachan Kim, Junho Kim, and SangKeun Lee. 2024. Towards robust and generalized parameter-efficient fine-tuning for noisy label learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, et al. 2024. T\ulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.
- Jiaxiang Li, Siliang Zeng, Hoi-To Wai, Chenliang Li, Alfredo Garcia, and Mingyi Hong. 2024. Getting more juice out of the sft data: Reward learning from human demonstration improves sft for llm alignment. *arXiv preprint arXiv:2405.17888*.
- Xize Liang, Chao Chen, Jie Wang, Yue Wu, Zhihang Fu, Zhihao Shi, Feng Wu, and Jieping Ye. 2024. Robust preference optimization with provable noise tolerance for llms. *arXiv preprint arXiv:2404.04102*.
- Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648. PMLR.
- Junyu Luo, Xiao Luo, Xiusi Chen, Zhiping Xiao, Wei Ju, and Ming Zhang. 2024. Semievolt: Semi-supervised fine-tuning for llm adaptation. *arXiv preprint arXiv:2410.14745*.

- Kaifeng Lyu, Haoyu Zhao, Xinran Gu, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. 2024. Keeping llms aligned after fine-tuning: The crucial role of prompt templates. *arXiv preprint arXiv:2402.18540*.
- Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Walenius, and Pyry Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796.
- Elena Merdjanovska, Ansar Aynetdinov, and Alan Akbik. 2024. NoiseBench: Benchmarking the impact of real label noise on named entity recognition. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.
- Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.
- Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawa-har, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.
- Jishnu Mukhoti, Yarin Gal, Philip HS Torr, and Puneet K Dokania. 2023. Fine-tuning can cripple your foundation model; preserving features may be the solution. *arXiv preprint arXiv:2308.13320*.
- Curtis Northcutt, Lu Jiang, and Isaac Chuang. 2021. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411.
- Ritik Sachin Parkar, Jaehyung Kim, Jong Inn Park, and Dongyeop Kang. 2024. Selectllm: Can llms select important instructions to annotate? *arXiv preprint arXiv:2401.16553*.
- Zhenting Qi, Xiaoyu Tan, Chao Qu, Yinghui Xu, and Yuan Qi. 2023. SaFER: A robust and efficient framework for fine-tuning BERT-based classifier with noisy labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*.
- Dan Qiao, Chenchen Dai, Yuyang Ding, Juntao Li, Qiang Chen, Wenliang Chen, and Min Zhang. 2022. Selfmix: Robust learning against textual label noise with self-mixup training. *arXiv preprint arXiv:2210.04525*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.
- Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608.
- Qi Sun, Kun Huang, Xiaocui Yang, Pengfei Hong, Kun Zhang, and Soujanya Poria. 2023. Uncertainty guided label denoising for document-level distant relation extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- 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.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size, 2024. <https://arxiv.org/abs/2408.00118>, 1(2):3.
- Ryan Teknium, Jeffrey Quesnelle, and Chen Guang. 2024. *Hermes 3 technical report*. Preprint, arXiv:2408.11857.
- Tao Tu, Anil Palepu, Mike Schaeckermann, Khaled Saab, Jan Freyberg, Ryutaro Tanno, Amy Wang, Brenna Li, Mohamed Amin, Nenad Tomasev, et al. 2024. Towards conversational diagnostic ai. *arXiv preprint arXiv:2401.05654*.
- Song Wang, Zhen Tan, Ruocheng Guo, and Jundong Li. 2023a. Noise-robust fine-tuning of pretrained language models via external guidance. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12528–12540.
- Tianlu Wang, Ilia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. 2024. Self-taught evaluators. *arXiv preprint arXiv:2408.02666*.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 13484–13508.
- Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. 2024. Assessing the brittleness of safety alignment via pruning and low-rank modifications. *arXiv preprint arXiv:2402.05162*.
- Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.

Pengyu Xu, Liping Jing, and Jian Yu. 2024. Enhancing multi-label text classification under label-dependent noise: A label-specific denoising framework. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5674–5688.

Bo Yuan, Yulin Chen, Yin Zhang, and Wei Jiang. 2024. Hide and seek in noise labels: Noise-robust collaborative active learning with llms-powered assistance. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10977–11011.

Yivan Zhang, Gang Niu, and Masashi Sugiyama. 2021. Learning noise transition matrix from only noisy labels via total variation regularization. In *International Conference on Machine Learning*, pages 12501–12512. PMLR.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. [Llamafactory: Unified efficient fine-tuning of 100+ language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Yuchen Zhuang, Yue Yu, Lingkai Kong, Xiang Chen, and Chao Zhang. 2023. Dygen: Learning from noisy labels via dynamics-enhanced generative modeling. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3674–3686.