Neural Networks 193 (2026) 107989

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Full Length Article

Pay more attention to the robustness of LLMs on adversarial prompt for instruction data mining

Qiang Wang, Dawei Feng *, Xu Zhang, Ao Shen, Yang Xu, Bo Ding, Huaimin Wang

National Key Laboratory of Parallel and Distributed Computing, College of Computer Science and Technology, National University of Defense Technology, Hunan Changsha, 410073, China

ARTICLE INFO

Keywords:
Instruction tuning
Instruction data mining
LLMs' Robustness
Adversarial instruction-following difficulty
Adversarial instruction output embedding consistency

ABSTRACT

Instruction tuning has emerged as a paramount method for tailoring the behaviors of LLMs. Recent studies have unveiled the potential for LLMs to achieve high performance through fine-tuning with a limited quantity of high-quality instruction data. Instruction-Following Difficulty is one of the most representative approaches in instruction data mining, which involves selecting samples where LLMs fail to generate response that align with the provided instructions as the high-quality instruction data. Building upon this approach, we further investigate how the robustness of LLMs to adversarial prompts influences the selection of high-quality instruction data. This paper proposes a pioneering framework of high-quality instruction data mining for instruction tuning, focusing on the impact of LLMs' robustness on adversarial prompts. Our notable innovation is to generate adversarial instruction data by attacking the prompts associated with instruction samples. Then, we introduce an Adversarial Instruction-Following Difficulty (AIFD) metric, which utilizes complete instruction sample pairs to identify samples with high adversarial instruction difficulty as high-quality instruction data. Apart from it, to address cases where LLM responses deviate from user intent, we further introduce a novel Adversarial Instruction Output Embedding Consistency (AIOEC) method that relies solely on instruction prompts to mine high-quality online instruction data. We conduct extensive experiments on two benchmark datasets to assess the performance. The experimental results serve to underscore the effectiveness of our proposed two methods. Moreover, the results underscore the critical practical significance of considering the robustness of LLMs on adversarial prompts for instruction data mining.

1. Introduction

Large Language Models (LLMs) (Achiam et al., 2023; Brown et al., 2020; Radford et al., 2018, 2019) often encounter significant challenges in accurately interpreting human intent, which can result in the generation of responses that are inappropriate or misaligned with the user's expectations (Bakker et al., 2022; Zhang et al., 2023). To tackle this challenge, instruction tuning has emerged as a paramount method for customizing the behaviors of LLMs. The core of Instruction tuning (Longpre et al., 2023; Ouyang et al., 2022; Peng et al., 2023; Shu et al., 2023) is to leverage meticulously crafted instruction data to direct LLMs in generating responses that are in line with human expectations.

However, the construction of instruction data usually requires experts to manually calibrate, and massive amounts of data requires high cost and resources (Liang et al., 2024; Yuan et al., 2024). Therefore, instruction data mining has emerged as a promising approach for selecting high-quality data to enhance the LLMs' performance. Recent studies found that LLMs can be fine-tuned to perform well even with a small amount of high-quality instruction data (Du et al., 2023; Zhou et al., 2023). The most representative method for instruction tuning is based on self-guided, denoting the Instruction-Following Difficulty (IFD) as the metric to select the high-quality instruction data for instruction tuning(Li et al., 2023b).

While prior work such as IFD has shown that LLMs are capable of self-assessing instruction-following difficulty to identify high-quality data without human supervision, these approaches often overlook a critical vulnerability of LLMs: their sensitivity to prompts. The order of few-shot examples, minor typos, or different expressions with the same semantic meaning can lead to entirely different results (Lu et al., 2022; Maus et al., 2023; Si et al., 2023; Wang et al., 2024). As depicted in Fig. 1, the synonymous substitutions can result in significant performance degradation in tasks like mathematical reasoning and sentiment

* Corresponding authors.
E-mail addresses: happytiger_95@163.com (Q. Wang), davyfeng.c@qq.com (D. Feng), zhangxu09@nudt.edu.cn (X. Zhang), shenaolgdx@sina.com (A. Shen), 1109618978@qq.com (Y. Xu), dingbo@nudt.edu.cn (B. Ding), hmwang@nudt.edu.cn (H. Wang).

https://doi.org/10.1016/j.neunet.2025.107989
Received 11 February 2025; Received in revised form 30 June 2025; Accepted 13 August 2025
Available online 22 August 2025

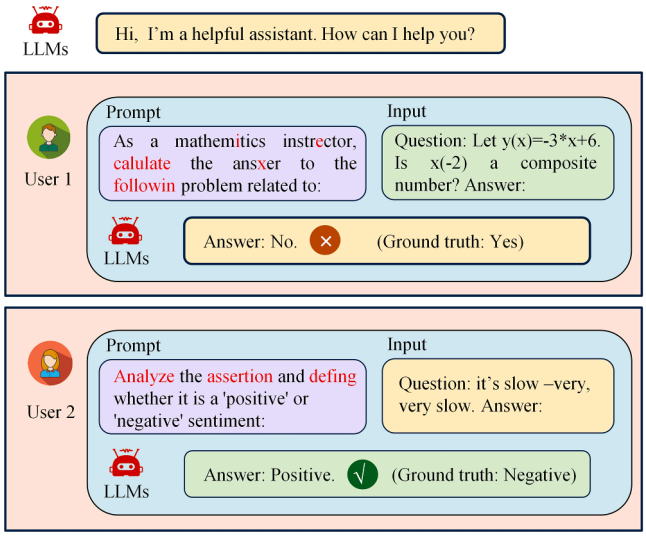


Fig. 1. Large Language Models (LLMs) exhibit a notable lack of robustness to prompt variations: even minor perturbations such as typographical errors or synonymous substitutions can result in significant performance degradation in tasks like mathematical reasoning and sentiment analysis. The red characters and words are perturbations.

analysis. This observation suggests that difficulty estimates based solely on static prompts may fail to reflect the true robustness of instruction data in real-world, noisy settings.

Motivated by this gap, we propose to evaluate not only the static quality of instructions, but also LLMs' adversarial robustness across diverse prompt perturbations. Our goal is to enhance the overall effectiveness of instruction tuning by selecting data that meets the criteria of instructional quality while maintaining adversarial robustness across diverse scenarios.

To thoroughly assess the adversarial robustness of prompts, we adopt a comprehensive suite of adversarial attack strategies spanning three linguistic granularity levels: character-level (Gao et al., 2018; Li et al., 2019), word-level (Jin et al., 2020; Li et al., 2020), and sentence-level (Naik et al., 2018; Ribeiro et al., 2021). The integration of these three categories ensures comprehensive coverage of prompt vulnerabilities, capturing both shallow and deep perturbation effects. Fig. 2 illustrates the overall framework of high-quality instruction data (named as diamond) mining from online instruction data, integrating two novel self-guided approaches to select diamond. We perform three different types of attacks for the user's prompt, to generate different adversarial instruction samples (Zhu et al., 2023). Based on the adversarial instruction samples and IFD algorithm, we propose a novel Adversarial Instruction-Following Difficulty (AIFD) score, a self-guided approach enabling models to autonomously select diamond data from instruction data. The higher AIFD score of instruction data, indicating less guidance and lower robustness, the more it should be mined for instruction tuning. And this approach is constrained by the necessity for the LLMs' responses in instruction data to conform to the human expectations.

However, during the online inference process, LLMs frequently exhibit a significant mismatch between their response and human expectations. As depicted in Fig 2, Case 1 demonstrates a successful interaction in which the model generates a response well-aligned with the user's intent. In contrast, Case 2 highlights a failure case where the model fails to generate a satisfactory response, thereby falling short of user expectations. In Case 2, applying the AIFD method mine high-quality data from online instruction data may result in the selected data not necessarily meeting the expected quality standards. Therefore, in this study, to address cases where LLM responses deviate from user intent, we further introduce a novel Adversarial Instruction Output Embedding Consistency (AIOEC) method that relies solely on instruction prompts to mine high-quality online instruction data.

We briefly summarize our contribution as following:

- We propose a novel framework for diamond data mining from instruction data. Considering the robustness of large to adversarial prompts influences the selection of high-quality instruction data.Recognizing that the robustness of LLMs to adversarial prompts critically affects the selection of high-quality instruction data, we introduce the Adversarial Instruction-Following Difficulty (AIFD) score as a metric to select the instruction data as diamond data.
- To address cases where LLM responses deviate from user intent, we further introduce a novel Adversarial Instruction Output Embedding Consistency (AIOEC) method that relies solely on instruction prompts to mine high-quality online instruction data.
- We conduct an extensive set of experiments on two benchmark datasets to assess the performance. The experimental results serve to underscore the effectiveness of our proposed two methods.

2. Related work

Numerous efforts have been made to enhance instruction tuning, which has become a pivotal approach for fine-tuning the behaviors of large language models (LLMs). Research in this area primarily focuses on two key aspects: (1) instruction data mining and (2) core set selection. The most notable works in this domain are summarized as follows:

2.1. Instruction data mining

While it is widely recognized that high-quality data plays a crucial role in instruction tuning, the exploration of non-human-curated sources of such data remains a largely untapped area. Instruction Mining (Cao et al., 2023) systematically evaluates a range of metrics and utilizes a statistical regression model for data selection, leveraging the training of multiple models to identify the most relevant data points. In contrast, ALPAGASUS (Chen et al., 2023) employs an external, fully-trained ChatGPT model to evaluate and score each sample. While this method proves effective, it may underutilize the inherent capabilities of the base model, placing excessive reliance on external models rather than fully leveraging the model's own strengths. The MoDS (Du et al., 2023) mines data based on three criteria: quality, coverage, and necessity. Quality ensures high-quality questions and answers, coverage guarantees diverse and comprehensive instructions, and necessity targets tasks that challenge large models, addressing their limitations. Superfiltering (Li et al., 2024) reduces data filtering time and cost by using smaller language models, such as GPT-2, in place of larger models. Selective Reflection-Tuning (Li et al., 2023a) leverages the teacher model to refine existing instruction-tuning data through reflection, while the student model selectively absorbs these improvements based on its own characteristics. This approach enhances performance without requiring new data, ensuring high sample efficiency.

2.2. Robustness-based selection

QDIT (Quality-Diversity Instruction Tuning) (Bukharin & Zhao, 2023) is a novel algorithm designed to optimize both the diversity and quality of instruction tuning datasets. By employing a greedy strategy, QDIT selects data points that most effectively enhance the joint quality-diversity score, significantly improving dataset efficiency. This approach is highly scalable, capable of handling datasets with millions of instructions. Research demonstrates that, while maintaining or improving best-case and average performance, the algorithm substantially enhances the robustness of the worst-case scenarios. Clustering and Ranking (CaR) (Ge et al., 2024) is a lightweight yet effective instruction data selection framework that combines expert-aligned quality scoring with diversity-preserving clustering. By leveraging a 550M-parameter scoring model and local clustering, CaR achieves high-quality coverage with minimal data, offering a practical solution for scalable instruction

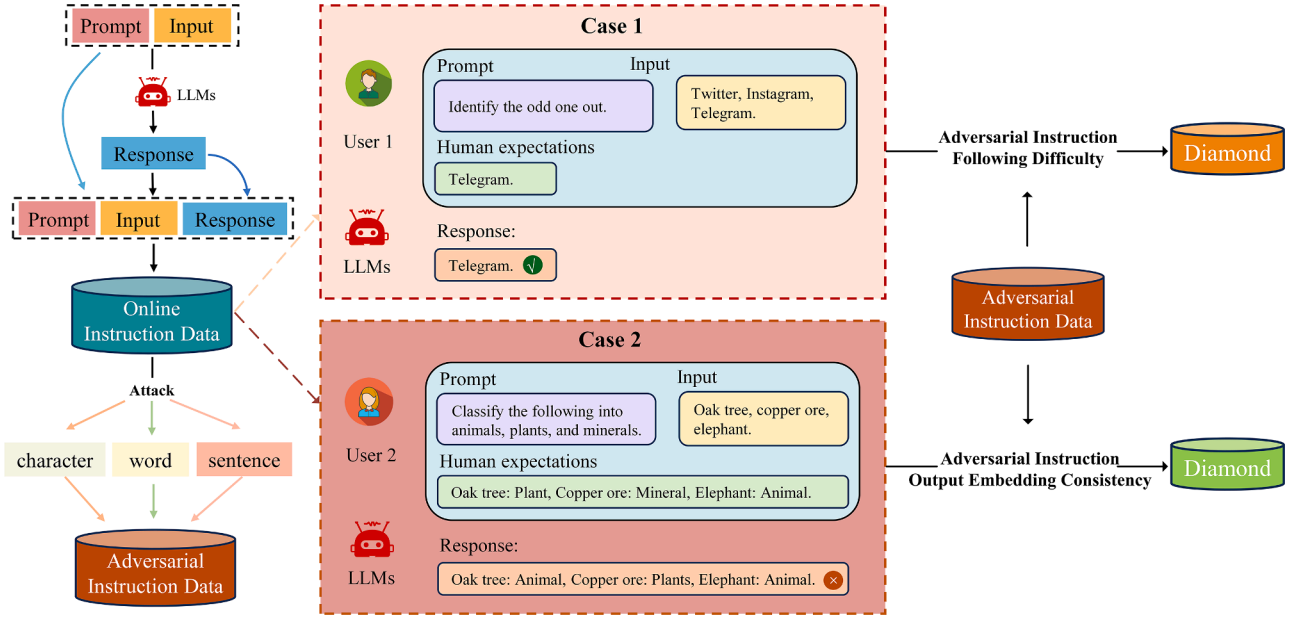


Fig. 2. The illustration of the overall framework of diamond mining from online instruction data to conduct the instruction tuning. The framework integrates two novel self-guided approaches to select diamond. We perform three different types of attacks, the character-level, word-level, and sentence-level for the user's prompt, to generate different adversarial instruction samples. Based on the adversarial instruction samples and IFD algorithm, we propose a novel Adversarial Instruction-Following Difficulty (AIFD) score, a self-guided approach enabling models to autonomously select diamond data from instruction data. Apart from it, we introduce a novel Adversarial Instruction Output Embedding Consistency as the metric to mine the diamond data, through measuring the output embedding similarity between the adversarial prompts and original prompt.

tuning. GPO (Generalized Prompt Optimization) leverages zero-shot labeling and prompt ensembling techniques of large language models to enable automated and reliable annotation across different data groups, which improves the performance on the target group while maintaining strong results on the source group. Compared to existing approaches, GPO is the first to systematically address the lack of quantitative analysis and methodological framework for prompt robustness. Sun et al. (2023) proposed a simple method to improve robustness by imposing an objective encouraging LLMs to induce similar representations for semantically equivalent instructions. This consistently improves the performance realized when using novel but appropriate task instructions.

Our work aims to introduce a novel methodology that harnesses the representation features of the target model to identify high-quality data for instruction tuning, with a particular focus on the LLMs' robustness on adversarial prompts. By emphasizing simplicity and efficiency, our approach offers a more effective and streamlined solution, driving the advancement of the field.

2.3. Coreset selection

The coreset selection is pivotal in machine learning, as it aims to identify a representative subset from massive datasets, thereby accelerating the learning process across various models. This approach has proven effective across a diverse set of machine learning algorithms, including adversarial contrastive learning (Xu et al., 2023), continual learning (Hao et al., 2023), updated Support Vector Machines (Tsang et al., 2005), K-means clustering (Har-Peled & Kushal, 2005), logistic regression (Munteanu et al., 2018), and semi-supervised learning (Killamsetty et al., 2021).

Recent advancements in neural network training, exemplified by Toneva et al. (Toneva et al., 2019), explore the dynamics of data point utility during training. Their findings reveal that points infrequently forgotten have minimal impact on final model accuracy. Paul et al. (2021) further demonstrate that averaging the expected loss gradient norm scores over different weight initializations effectively prunes training data without significantly sacrificing accuracy. Mindermann et al. (2022) utilize Bayesian probability theory to assess the individual contribution of each training point to the holdout loss, thereby enabling more precise control over the training process and significantly improving training efficiency. Wan et al. (2024) extend the concept of final similarity by incorporating the differences between dimensions, thereby improving the selection process and enhancing the overall effectiveness of the core set. Additionally, Xia et al. (2023) investigate the robustness of the core set across various scenarios, considering how its performance adapts to changing conditions.

3. Methodology

3.1. Background for instruction-following difficulty

During the instruction tuning phase, the loss of a sample pair (Q, A) is computed through continuously predicting the subsequent tokens, considering both the instruction Q and their proceeding words in the sequence:

$$L_\theta(A|Q) = -\frac{1}{N}\sum_{i=1}^{N} log P(w_i^A|Q, w_1^A, w_2^A, \ldots, w_{i-1}^A; \theta) \quad (1)$$

where N is the number of words in the ground-truth answer A, and instruction Q encompasses both the prompt P and the input I. In addition, this averaged cross-entropy loss is referred as the conditioned answer score, denoted as $s_\theta(A|Q) = L_\theta(A|Q)$. This metric assessed the model's ability to generate an appropriate response according to given instruction Q.

However, the $s_\theta(A|Q)$ does not account for the inherent difficulty of aligning A itself. Therefore, to better estimate the difficulty of following given sample instructions, a direct answer score $s_\theta(A)$ to measure the LLMs' ability to generate the answer independently is introduced as following:

$$s_\theta(A) = -\frac{1}{N}\sum_{i=1}^{N} \log P(w_1^A, w_2^A, \ldots, w_{i-1}^A; \theta) \quad (2)$$

And this score quantifies the inherent difficulty associated with autonomously generating an answer without explicit instructions. A higher score in the direct answer metric signifies a heightened level of difficulty in answer generation by the model.

To obtain better instruction data, identifying which instructions have a greater impact on the model while excluding the influence of the answers themselves, the concept of Instruction-Following Difficulty (IFD) score is proposed as follows:

$$r_\theta(Q, A) = \frac{s_\theta(A|Q)}{s_\theta(A)} \tag{3}$$

Utilizing the IFD metric for data filtering has mitigated the impact of large models on fitting the answers themselves, allowing for a direct assessment of the influence of given instructions on the model's answer generation. Higher IFD scores indicate the model's inability to align the answer with the provided instructions, highlighting greater difficulty in instruction comprehension and presenting an advantageous opportunity for model refinement.

3.2. Adversarial instruction following difficulty

To thoroughly assess the adversarial robustness of prompts, we adopt a comprehensive suite of adversarial attack strategies spanning three linguistic granularity levels: character-level, word-level, and sentence-level. In addition, the different types of attach processes are as follows:

Character-level: We utilize the TextBugger (Li et al., 2019) and DeepWordBug (Gao et al., 2018), both of which manipulate texts by introducing typos or errors into words. This manipulation can include adding, deleting, repeating, replacing, and permuting characters within certain words.

Word-level: We employ the BertAttack (Li et al., 2020) and TextFooler (Jin et al., 2020), sophisticated techniques designed to replace words with synonyms or contextually similar alternatives, with the goal of deceiving large language models.

Sentence-level: We deploy the StressTest (Naik et al., 2018) and CheckList (Ribeiro et al., 2021) methodologies, both of which involve appending irrelevant or extraneous sentences to prompts, aiming to distract LLMs. For example, in the CheckList attack, we generate 50 random sequences of alphabets and digits. Table 1 depicts the example of adversarial prompts generated by 6 attacks.

The integration of these three categories ensures comprehensive coverage of prompt vulnerabilities, capturing both shallow and deep perturbation effects. This systematic approach allows us to approximate a robustness-aware data mining process that not only emphasizes difficulty in alignment but also resilience to prompt variation-a key aspect when deploying instruction-tuned models in the open world.

Our core motivation is to minimize cross-entropy loss during the model inference process, via the adversarial instruction data. In this paper, we propose the Adversarial Instruction Following Difficulty score to select high-quality data, a metric devised to evaluate the difficulty each adversarial instruction data presents.

Table 1
Example of adversarial prompts generated by 6 attacks.

Clean	Give three tips for staying healthy.
TextBugger	Give three tips for staying helthy.
DeepWordBug	give three tips for staying healthy.
TextFooler	Give three tips for staying salubrious.
BertAttack	Give three counseling for staying healthy.
CheckList	Give three tips for staying healthy zq0DcZ5dnI.
StressTest	Give three tips for staying healthy and false is not true.

Specially, we try to estimate the **Adversarial Instruction-Following Difficulty (AIFD)** scores $r_\theta(Q, A)$ on following adversarial instruction of six different given (Q_A, A)pairs by calculating the ratio between $s_\theta(A)$ and $s_\theta(A|Q_A)$:

$$r_\theta(Q, A, Q_A) = \frac{s_\theta(A|Q)}{s_\theta(A)} + \frac{1}{6}\left(\sum_{i=1}^{i=6} \frac{s_\theta(A|Q_A^i)}{s_\theta(A)}\right), Q_A^i \in Q_A \tag{4}$$

By leveraging this score, the influence of LLMs' intrinsic ability to fit the answer string is partially alleviated. The score measures the degree to which given adversarial instruction benefits the alignment of corresponding response. High AIFD scores infer the inability of the model to align response to the given corresponding adversarial instructions. which in turn indicates the difficulty of an adversarial instruction. Algorithm 1 describes the diamond data mining process from online instruction data via the AIFD approach. **Calibration in** Algorithm 1 involves using GPT-4 to correct and refine the LLMs' outputs in response to online instructions, during the online inference phase, when responses to user inputs may occasionally be incorrect or less accurate. In this paper, We employ a multi-granular adversarial attack framework, systematically perturbing prompts at three distinct linguistic levels.

3.3. Adversarial instruction output embedding consistency

The AIFD method evaluates the quality of instruction data through the instruction and response pairs. However, during the inference process, LLMs often generate outputs that diverge significantly from human expectations. Due to the unreliable or suboptimal response, the resulting inconsistency compromises the reliability of the response component, thereby limiting the applicability of AIFD for online instruction data evaluation.

To address this limitation, we propose a novel approach-Adversarial Instruction Output Embedding Consistency (AIOEC), which assesses the quality of instruction data based solely on the prompt, independent of the potentially noisy response. The AIOEC framework for diamond data mining is illustrated in Fig. 3. The method consists of three main steps:

Algorithm 1 Diamond data mining from online instruction data via the AIFD approach.

Input: Online prompt set $\{P_1, P_2, \ldots, P_N\}$, corresponding input set $\{I_1, I_2, \ldots, I_N\}$, total number of instruction samples N, LLMs $\mathcal{T}_{\text{LLM}}$
Output: High-quality (diamond) instruction data $\mathcal{D}_{\text{diamond}}$

for $n = 1$ to N **do**
  $A_n = \mathcal{T}_{\text{LLM}}(P_n, I_n)$ ▷ Generate initial response
  $\mathcal{D}_n = \text{Construct}(P_n, I_n, A_n)$ ▷ Form the base instruction data
  $P_n^{\text{adv}} = \text{Attack}(P_n)$ ▷ Generate adversarial prompt
  $\mathcal{D}_n^{\text{adv}} = \text{Construct}(P_n^{\text{adv}}, I_n, A_n)$ ▷ Form adversarial instruction data
  $S_n = \text{AIFD}(\mathcal{D}_n^{\text{adv}}, \mathcal{D}_n)$ ▷ Evaluate instruction difficulty
end for
$\{\mathcal{D}_n\} = \text{SortBy}(S_n)$ ▷ Sort instruction data by AIFD score
$\mathcal{D}_{\text{core}} = \text{SelectTopK}(\{\mathcal{D}_n\})$ ▷ Select core instruction data subset
$\mathcal{D}_{\text{diamond}} = \text{Calibrate}(\mathcal{D}_{\text{core}})$ ▷ Calibrate responses by GPT-4
return $\mathcal{D}_{\text{diamond}}$

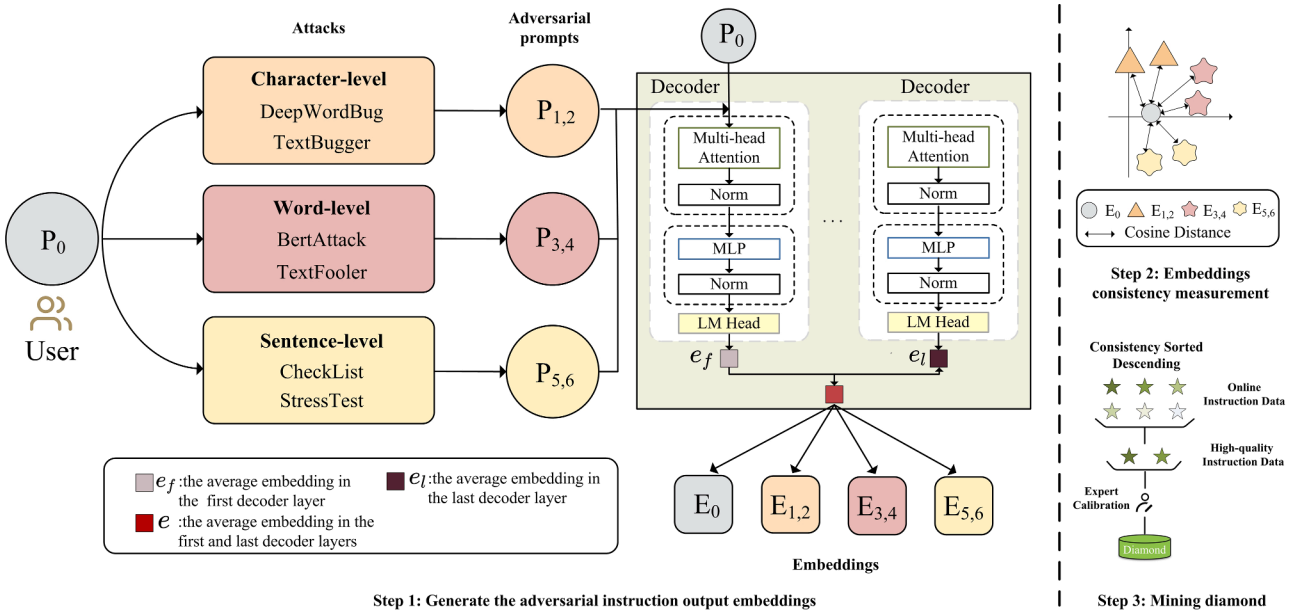


Fig. 3. The illustration of Adversarial Instruction Output Embedding Consistency for diamond data mining. The AIOEC method consists of three steps. First, we generate six adversarial prompts by conducting character-level, word-level, and sentence-level attacks on the user's input prompt. These adversarial prompts, along with the original user prompt, are then fed into the LLMs to generate corresponding output embeddings. The second step involves evaluating the consistency between the output embeddings E_A from the adversarial prompts and E from the user's prompt. Finally, a selected proportion of the highest-ranked data is chosen as diamond data for instruction tuning.

Step 1: Prompt perturbation and embedding generation.. Given a user-provided instruction prompt, we generate six adversarial variants using character-level, word-level, and sentence-level attacks. These adversarial prompts, along with the original prompt, are then fed into the LLM to obtain the corresponding output embeddings.

Step 2: Embedding consistency assessment.. We evaluate the consistency between the output embedding E of the original prompt and the set of embeddings $E_A = \{E_A^1, E_A^2, \ldots, E_A^6\}$ generated from its adversarial variants. Each embedding is computed as the sum of two components:

- e_f: the average of all token embeddings from the first hidden layer of the LLM,
- e_l: the average of all token embeddings from the last hidden layer of the LLM.

Thus, the final embedding is defined as:

$$E = e_f + e_l \tag{5}$$

The overall consistency score is calculated as the average cosine similarity between the original output embedding and its adversarial counterparts:

$$\mathrm{dis}(E, E_A) = \frac{1}{6}\sum_{i=1}^{6} \cos(E, E_A^i), \quad E_A^i \in E_A \tag{6}$$

Step 3: Instruction quality ranking and selection.. Based on the computed consistency score, all candidate instructions are ranked in descending order. A predefined top proportion is selected as diamond-quality instruction data, which is subsequently used for instruction fine-tuning.

4. Experiments

4.1. Experimental setting

Foundation large language model. To demonstrate the effectiveness of our proposed framework, we conducted extensive testing on four models from two widely recognized open-source large language models families, including LLaMA-7B (Touvron et al., 2023a), LLaMA2-7B (Touvron et al., 2023b), Mistral-7B-v0.1 (Jiang et al., 2023) and Mistral-7B-v0.3.

Evaluation and datasets. To assess the performance of the model for instruction tuning, we follow LLaMA's evaluation to perform zero-shot task classification on common sense reasoning datasets: MMLU (Hendrycks et al., 2021), HellaSwag (Zellers et al., 2019), ARC (Clark et al., 2018), TruthfulQA (Lin et al., 2022) on the OpenCompass assessment framework (Contributors, 2023).

Training datasets. The Alpaca datasets (Taori et al., 2023) encompass 52,002 instruction-following samples. WizardLM dataset (Xu et al., 2023) leverages the EvoInstruction data. We utilize the WizardLM70K for our experiment.

4.2. Implementation details

In our experiments, we fine-tuned the pre-trained LLMs by utilizing the Diamond data, employing a training configuration consistent with the original Alpaca and WizardLM, utilizing the llm-action codebase. During the experiments, the model was trained using the Adam optimizer with a batch size of 128, spanning 3 epochs for training. The original learning rate was 0.00002, and the warmup rate was 0.03. For experiments conducted on the pre-trained LLaMA-7B model, our training framework adheres to protocols similar to those used in the Alpaca and WizardLM datasets. When training on the Alpaca dataset, a maximum input length of 512 was employed. And we opted for a 1024 input length when using the WizardLM dataset.

In experiments involving the LLaMA2-7B, Mistral-7B-v0.1, Mistral-7B-v0.3, we employed instruction prompts derived from Vicuna. And in this paper, all experiments are based on the eight V100 graphics cards and the PyTorch 2.1 framework. We conducted five seed experiments and selected the median of their results as the final outcome.

4.3. Experimental results and analysis

4.3.1. The diamond data mining from offline instruction datasets

To demonstrate the effectiveness of our approach, which emphasizes the significance of considering the prompts' adversarial robustness in

Table 2
The comparison of performance for fine-tuned LLaMA-7B and LLaMA2-7B incorporating Alpaca datasets of different proportions as diamond data on different tasks.

Datasets	Methods	LLaMA-7B					LLaMA2-7B				
		ARC	HellaSwag	MMLU	TruthfulQA	Average	ARC	HellaSwag	MMLU	TruthfulQA	Average
	Pre-train	34.27	74.31	35.53	34.31	44.61	50.47	74.00	46.66	38.96	52.52
5 % Alpaca	IFD	36.80	76.60	37.05	36.92	46.84	38.29	76.65	46.00	47.11	52.01
	AIFD	**38.43**	**76.72**	**37.54**	**39.45**	**48.04**	**39.17**	**76.89**	**46.65**	**47.47**	**52.55**
10 % Alpaca	IFD	37.57	76.56	36.75	37.41	47.07	37.48	76.47	45.45	46.70	51.53
	AIFD	**40.10**	**76.64**	**37.96**	**38.73**	**48.36**	**40.64**	**76.60**	**45.55**	**47.28**	**52.52**
15 % Alpaca	IFD	37.40	77.45	35.18	38.05	47.02	43.74	76.40	47.18	48.02	53.84
	AIFD	**42.18**	**78.49**	**36.22**	**39.23**	**49.03**	**48.71**	**76.72**	**47.54**	**48.61**	**55.40**
20 % Alpaca	IFD	38.38	79.24	38.60	39.69	48.98	44.93	76.47	47.22	46.56	53.80
	AIFD	**38.78**	**79.46**	**40.42**	**41.05**	**49.93**	**45.67**	**76.60**	**47.41**	**46.50**	**54.00**
Official Alpaca		42.48	76.91	42.16	39.55	50.25	50.23	77.65	46.75	44.87	54.88

Table 3
The comparison of performance of LLaMa-7B fine-tuned with WizardLM70K data at various ratios for different tasks.

Datasets	Methods	ARC	HellaSwag	MMLU	TruthfulQA	Average
10 % WizardLM70K	IFD	37.56	75.35	36.25	43.89	48.26
	AIFD	**41.62**	**75.69**	**37.18**	**45.23**	**49.93**
20 % WizardLM70K	IFD	43.58	74.82	37.78	44.63	50.20
	AIFD	**44.82**	**74.93**	**38.57**	**44.62**	**50.74**
40 % WizardLM70K	IFD	46.03	73.38	39.57	45.04	51.00
	AIFD	**47.62**	**73.55**	**40.14**	**45.65**	**51.74**

facilitating diamond data mining, we firstly conducted experiments on the offline instruction datasets.

Table 2 illustrates the comparison of performance for fine-tuned LLaMA-7B and LLaMA2-7B incorporating Alpaca data as diamond data across for different tasks, via the opencompass evaluation tool. According to Table 2, we can observe that our proposed AIFD approach achieves **1.2 %** higher average accuracy than IFD approach on 4 different tasks, only utilizing 5 % diamond data to fine-tune the LLaMA-7B model. After fine-tuning with diamond data mined via our AIFD approach, the LLaMA-7B's performance increased by **4.43 %** compared to the pre-trained LLaMA-7B model, slightly lower than the LLaMA-7B with official Alpaca, which was **2.21 %**. In addition, the experimental results show that the performance improvement of the LLaMA2-7B fine-tuned with 5 % diamond data based on our AIFD method compared to IFD is **0.54 %**.

Furthermore, we curate subsets comprising the top 10 %, 15 % and 20 % of the Alpaca datasets as the diamond data to fine-tune the LLaMA-7B model, facilitating an investigation into performance fluctuations. As depicted in Table 2, we present the performance comparison of LLaMA-7B fine-tuned at various proportions of Alpaca datasets. A consistent finding emerges: LLaMA-7B consistently outperforms the IFD method when fine-tuning with data from the Alpaca datasets using our proposed AIFD approach. In addition, the experimental results show that we can select more high-quality instruction data for instruction tuning, via our AIFD method, considering the impact of prompts' adversarial robustness on instruction data.

Moreover, we conducted instruction tuning experiments on the WizardLM70K dataset. We craft subsets containing the top 10 %, 20 % and 40 % of the WizardLM70K datasets as the diamond data to fine-tune the LLaMA-7B model, enabling us to investigate the performance changes. Table 3 depicts the comparison of performance of LLaMa-7B fine-tuned with WizardLM70K Data at various ratios across different tasks. According to Table 3, we can observe that our proposed AIFD approach achieves **1.67 %** higher average accuracy than IFD approach on 4 different tasks, only utilizing 10 % diamond data to fine-tune the LlaMA-7B model.

To facilitate a more comprehensive comparison of our method with state-of-the-art (SOTA) approaches, we incorporate the MoDS (Du et al., 2023), Superfiltering (Li et al., 2024), Reflection-Tuning (Li et al., 2023a) and CaR, as baseline methods on Alpaca datasets. The experiments are conducted with LLaMA-7B and LLaMA2-7B. The comparison of performance for fine-tuned LLaMA-7B and LLaMA2-7B incorporating 5 % Alpaca datasets as diamond data on different tasks is depicted in Table 4. Our method has a clear advantage over the SOTA methods.

To better demonstrate the effectiveness of our method and to account for architectural diversity, we also conducted the experiments on Mistral-7B-v0.1 and Mistral-7B-v0.3 from the Mistral family. Table 5 represents the comparison of performance for fine-tuned Mistral-7B-v0.1 and Mistral-7B-v0.3 incorporating 5 % Alpaca datasets as diamond data on different tasks. Our approach demonstrates a distinct superiority compared to the IFD method.

4.3.2. The diamond data mining from online instruction datasets

We delve further into exploring the efficacy of our proposed methods in more challenging scenario, specifically in selecting high-quality data from online instruction data. In addition, the comprehensive procedure for generating online command data can be located in Appendix A. We utilized AIFD and AIOEC methods to mine high-quality from online instruction data. Subsequently, we replaced the responses in this high-quality data with the real answers corresponding to the instructions from the offline data, for the purpose of instruction tuning.

Table 6 compares the performance for fine-tuned LLaMA-7B and LLaMA2-7B incorporating online Alpaca data of different proportions as diamond data on different tasks. Using our proposed methods, we selected 5 % of the online data as diamond data for fine-tuning LLaMA-7B and LLaMA2-7B. Both models show superior performance compared to the baseline IFD method.

Furthermore, we curate subsets comprising the top 10 %, 15 % and 20 % of the online Alpaca datasets as the diamond data to fine-tune the LLaMA-7B and LLaMA2-7B models. As depicted in Table 2, we present the performance comparison of LLaMA-7B and LLaMA2-7B fine-tuned at various proportions of Alpaca datasets. Both models show superior performance compared to the baseline IFD method.

Fig. 4 presents the comparison of performance for the fine-tuned LLaMA2-7B utilizing 10 % and 20 % WizarrdLM70K Datasets on the different tasks. And the results demonstrate that our proposed method can mine more high-quality instruction data.

4.4. Ablation study

To further investigate how the robustness of LLMs to adversarial prompts affects the selection of high-quality instruction data, we conducted additional training and evaluation on our proposed diamonds

Table 4
The comparison of performance with the SOTA method for the fine-tuned LLaMA-7B and LLaMA2-7B incorporating 5 % Alpaca datasets as diamond data on different tasks.

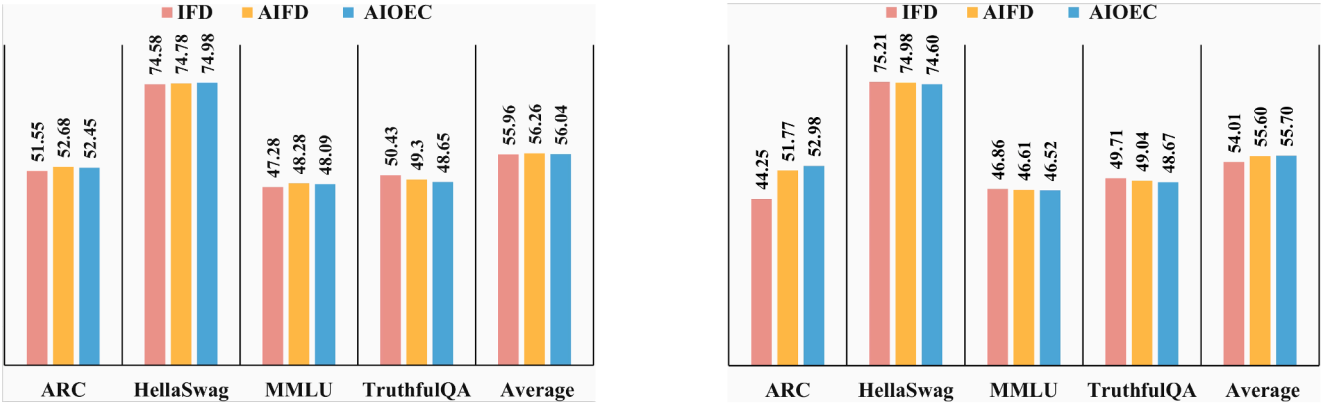
Datasets	Methods	LLaMA-7B					LLaMA2-7B				
		ARC	HellaSwag	MMLU	TruthfulQA	Average	ARC	HellaSwag	MMLU	TruthfulQA	Average
5 % Alpaca	MoDS	33.06	76.73	39.92	37.33	46.62	47.25	75.52	47.90	43.49	52.01
	Superfiltering	34.44	77.01	35.55	35.02	45.51	41.71	76.02	46.21	45.12	52.27
	Reflection-Tuning	37.14	77.10	36.99	37.38	47.15	39.49	75.62	45.72	44.76	51.40
	CaR	31.85	70.21	33.80	33.44	42.33	43.08	76.26	45.68	42.27	51.8
	IFD	36.80	76.60	37.05	36.92	46.84	38.29	76.65	46.00	47.11	52.01
	AIFD	**38.43**	**76.72**	**37.54**	**39.45**	**48.04**	**39.17**	**76.89**	**46.65**	**47.47**	**52.55**

Table 5
The comparison of performance for the fine-tuned Mistral-7B-v0.1 and Mistral-7B-v0.3 incorporating 5 % Alpaca datasets as diamond data on different tasks.

Datasets	Methods	Mistral-7B-v0.1					Mistral-7B-v0.3				
		ARC	HellaSwag	MMLU	TruthfulQA	Average	ARC	HellaSwag	MMLU	TruthfulQA	Average
5 % Alpaca	IFD	53.13	75.11	50.04	40.56	54.71	63.13	73.77	52.49	38.79	57.05
	AIFD	**59.88**	**75.12**	**51.38**	**41.26**	**56.91**	**62.42**	**73.86**	**56.14**	**39.20**	**57.91**

Table 6
The comparison of performance for fine-tuned LLaMA-7B and LLaMA2-7B incorporating online Alpaca data of different proportions as diamond data on different tasks.

Datasets	Methods	LLaMA-7B					LLaMA2-7B				
		ARC	HellaSwag	MMLU	TruthfulQA	Average	ARC	HellaSwag	MMLU	TruthfulQA	Average
5 % Alpaca	IFD	36.54	76.08	37.02	38.08	46.93	40.64	74.79	44.27	45.80	51.38
	AIFD	**39.76**	**75.82**	**36.68**	**38.62**	**47.72**	**50.29**	**73.99**	**46.85**	**43.64**	**53.69**
	AIOEC	**37.36**	**76.74**	**36.96**	**38.37**	**47.63**	**42.32**	**74.65**	**46.55**	**44.03**	**51.89**
10 %Alpaca	IFD	38.09	76.31	37.69	38.35	47.61	41.34	74.98	45.04	45.82	51.80
	AIFD	**38.58**	**76.57**	**38.26**	**39.16**	**48.14**	**51.88**	**74.76**	**47.25**	**44.63**	**54.63**
	AIOEC	**40.93**	**76.46**	**36.52**	**38.08**	**48.00**	**49.75**	**74.82**	**47.44**	**44.24**	**54.06**
15 % Alpaca	IFD	40.23	74.78	38.14	39.78	48.23	42.33	75.54	46.14	46.44	52.61
	AIFD	**44.43**	**75.67**	**38.68**	**40.55**	**49.83**	**53.31**	**75.19**	**48.11**	**44.81**	**55.36**
	AIOEC	**45.00**	**76.21**	**37.18**	**38.90**	**49.32**	**55.55**	**75.07**	**47.32**	**44.44**	**55.60**
20 % Alpaca	IFD	43.80	75.02	37.85	40.67	49.34	42.36	75.33	45.44	45.13	50.07
	AIFD	**45.01**	**75.78**	**39.45**	**40.16**	**50.10**	**57.31**	**75.45**	**47.78**	**44.90**	**56.36**
	AIOEC	**46.42**	**75.95**	**38.21**	**39.64**	**50.01**	**53.96**	**74.98**	**47.21**	**41.69**	**54.46**



(a) 10% WizardLM70K (b) 20% WizardLM70K

Fig. 4. The comparison of performance for the fine-tuned LLaMa2-7B utilizing 10 % and 20 % WizarrdLM70K datasets on the different tasks.

data mining framework. Building upon the baseline IFD method, we incorporate various types of attacks, and then select the top 5 % of instruction data as the diamond data for fine-tuning LLaMA2-7B, via the AIFD score.

The performance evaluation results of the fine-tuned LLaMA2-7B are presented in Table 7. According to the experimental results in Table 7, it can be observed that incorporating various attacks on the foundational IFD method yields consistent model performance. This indicates that introducing various types of attacks yields the same diamond data when filtered using the AIFD method. More explanation can be founded in Appendix C. It also suggests that different types of attacks have an equivalent effect on data selection. More importantly, we found

Table 7
The comparison of performance for the fine-tuned LLaMA2-7B utilizing 5 % Alpaca data on the different tasks by different attacks.

	ARC	HellaSwag	MMLU	TruthfulQA	Average
Three attacks	43.89	75.16	46.52	45.30	52.72
Character attack	39.95	72.31	44.84	39.09	49.05
Word attack	39.91	75.77	46.14	45.01	51.71
Sentence attack	38.83	72.84	45.19	39.77	49.16
IFD + Character	39.17	76.74	46.67	47.47	52.51
IFD + Word	39.17	76.74	46.67	47.47	52.51
IFD + Sentence	39.17	76.74	46.67	47.47	52.51

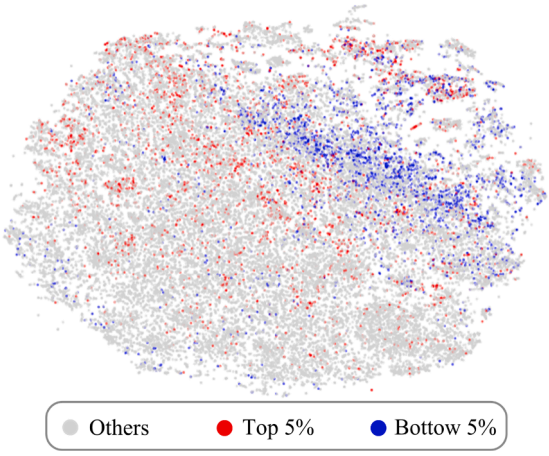


Fig. 5. Visualization using t-SNE on instruction embeddings from the Alpaca dataset.

that fine-tuning LLaMA2-7B with instruction data mined from just three types of attacks resulted in the best model performance.

4.5. The reliability analysis of AIFD

To better demonstrate the reliability of our method, we added some additional experiments. To generate poor-quality data, we used the base models (LLaMA-7B and LLaMA2-7B) combined with the trained LoRA as our online model, and then collected online Alpaca-LORA data. Based on the online Alpaca-LORA datasets, we conducted diamond data mining experiments based our method. The performance of the model obtained by combining the base model with LoRA tends to be subpar, and the quality of the generated responses is often low. Therefore, this experimental design is justified. For the online Alpaca-LORA dataset, we compared our method with IFD, IFD + Calibrate, Calibrate + AIFD. The method for Calibrate + AIFD involves using GPT-4 to calibrate the responses of all the online Alpaca-LORA data. Table 8 represents the experimental results. The experimental results demonstrate the reliability of our method.

Table 9
The top 10 occurred verb-noun pairs from the top5 % AIFD scores data and the low5 % AIFD scores data.

Top 5 % AIFD			Low 5 %AIFD		
Verb	Noun	Count	Verb	Noun	Count
Write	story	35	make	sentence	88
Generate	story	24	Rewrite	sentence	78
Write	essay	22	Edit	sentence	56
Generate	list	14	Change	sentence	29
Write	article	13	be	sentence	29
Create	story	12	use	sentence	25
Write	poem	12	make	text	20
Write	importance	11	go	sentence	19
Write	post	11	Take	sentence	19
Generate	word	10	correct	sentence	18

4.6. Diamond data characteristics

Distribution characteristics. In this section, our primary focus lies in comprehensively understanding the distributional characteristics of the diamond data within the original dataset. Initially, we calculate the embedding of each instruction in the Alpaca dataset using the pre-trained LLaMA-7B model. Subsequently, we employ t-SNE (Van der Maaten & Hinton, 2008) for dimensionality reduction, thereby mapping high-dimensional embeddings to a 2D space. The visualized vectors, color-coded according to the top or bottom 5 % difficulty ratios, are displayed in Fig. 5. Contrary to conventional beliefs, our carefully selected instruction exhibits non-uniform dispersion. Instead, discernible boundaries emerge between samples of high and low difficulty, challenging prior assumptions that selected data should span the entire spectrum of instructions and maximize diversity.

Pattern characteristics. To delve deeper into the pattern characteristics of the diamond data under scrutiny, we utilized the Berkeley Neural Parser (Kitaev & Klein, 2018). This sophisticated tool enables us to effectively delineate the verb-noun structure inherent in the instructions accompanying each data sample. Through this analytical framework, we are able to pinpoint the principal verb along with its corresponding direct noun object in each instruction. Consequently, this approach affords us direct insights into the types of instructions predisposed to receiving higher or lower AIFD scores. Our experimental methodology draws upon the Alpaca dataset, wherein we examine the top 10 verb-noun pairs derived from the highest and lowest 5 % AIFD scores, as presented in Table 9.

5. Conclusion

This study demonstrates the potential of leveraging adversarial prompts to mine high-quality instruction data that exhibits strong alignment with large language models. In this study, we propose a novel framework designed for online instruction data mining, integrating AIFD and AIOEC approaches. We conducted extensive experiments

Table 8
The comparison of performance for fine-tuned LLaMA-7B and LLaMA2-7B incorporating Alpaca-LORA datasets of different proportions as diamond data on different tasks.

Datasets	Methods	LLaMA-7B					LLaMA2-7B				
		ARC	HellaSwag	MMLU	TruthfulQA	Average	ARC	HellaSwag	MMLU	TruthfulQA	Average
5 % Alpaca-LORA	IFD	32.75	68.84	30.70	36.40	42.17	36.12	68.82	42.92	43.64	47.88
	IFD + Calibrate	33.97	70.10	32.33	35.45	42.96	42.45	75.63	41.80	44.70	51.15
	AIFD	**39.73**	**75.82**	**35.84**	**35.95**	**46.84**	**43.94**	**75.35**	**46.08**	**45.11**	**52.62**
	Calibrate + AIFD	**37.43**	**75.72**	**35.54**	**39.45**	**47.04**	**39.17**	**76.89**	**46.65**	**47.47**	**52.55**

on Alpaca and WizarrdLM-70k datasets, revealing the superior performance of the proposed approaches. Moreover, the results underscore the critical practical significance of considering LLMs' robustness.

To the best of our knowledge, this is the first investigation into the impact of LLMs' robustness for online instruction mining. We only analyze the impact of LLMs' robustness on high-quality data selection through generating adversarial instruction data by attacking prompts. We aspire this paper to inspire fellow researchers to delve into the LLMs' robustness for online instruction mining.

CRediT authorship contribution statement

Qiang Wang: Writing – original draft, Software, Methodology, Formal analysis, Conceptualization; **Dawei Feng:** Writing – review & editing, Validation; **Xu Zhang:** Visualization, Investigation; **Ao Shen:** Visualization, Validation; **Yang Xu:** Visualization, Validation; **Bo Ding:** Visualization, Validation; **Huaimin Wang:** Visualization, Validation.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by National Science and Technology Major Project (2023ZD0121101), the Open Fund of National Key Laboratory of Parallel and Distributed Computing (PDL) NO.2024KJWPDL-02 and National Key Laboratory under grant 231-HF-D04-01.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.neunet.2025.107989

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S. et al. (2023). GPT-4 technical report. arXiv preprint arXiv:2303.08774.

Bakker, M., Chadwick, M., Sheahan, H., Tessler, M., Campbell-Gillingham, L., Balaguer, J., McAleese, N., Glaese, A., Aslanides, J., Botvinick, M. et al. (2022). Fine-tuning language models to find agreement among humans with diverse preferences. *Advances in Neural Information Processing Systems*, *35*, 38176–38189.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodei, D. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (pp. 1877–1901). Curran Associates, Inc. (*Vol. 33*). https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Bukharin, A., & Zhao, T. (2023). Data diversity matters for robust instruction tuning. arXiv preprint arXiv:2311.14736.

Cao, Y., Kang, Y., & Sun, L. (2023). Instruction mining: High-quality instruction data selection for large language models. arXiv preprint arXiv:2307.06290.

Chen, L., Li, S., Yan, J., Wang, H., Gunaratna, K., Yadav, V., Tang, Z., Srinivasan, V., Zhou, T., Huang, H. et al. (2023). AlpaGasus: Training a better Alpaca model with fewer data. In *The twelfth international conference on learning representations*.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., & Tafjord, O. (2018). Think you have solved question answering? Try ARC, the AI2 reasoning challenge. arXiv preprint arXiv:1803.05457.

Contributors, O. (2023). OpenCompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

Du, Q., Zong, C., & Zhang, J. (2023). MoDS: Model-oriented data selection for instruction tuning. arXiv preprint arXiv:2311.15653.

Gao, J., Lanchantin, J., Soffa, M. L., & Qi, Y. (2018). Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE security and privacy workshops (SPW)* (pp. 50–56). IEEE.

Ge, Y., Liu, Y., Hu, C., Meng, W., Tao, S., Zhao, X., Ma, H., Zhang, L., Chen, B., Yang, H. et al. (2024). Clustering and ranking: Diversity-preserved instruction selection through expert-aligned quality estimation. arXiv preprint arXiv:2402.18191.

Hao, J., Ji, K., & Liu, M. (2023). Bilevel coreset selection in continual learning: A new formulation and algorithm. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in neural information processing systems* (pp. 51026–51049). Curran Associates, Inc. (*vol. 36*). https://proceedings.neurips.cc/paper_files/paper/2023/file/a0251e494a7e75d59e06d37e646f46b7-Paper-Conference.pdf.

Har-Peled, S., & Kushal, A. (2005). Smaller coresets for K-median and K-means clustering. In *Proceedings of the twenty-first annual symposium on computational geometry* (pp. 126–134).

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2021). Measuring massive multitask language understanding. In *9th International conference on learning representations, ICLR 2021, virtual event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=d7KBjmI3GmQ.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las, C. D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L. et al. (2023). Mistral 7B. arXiv preprint arXiv:2310.06825.

Jin, D., Jin, Z., Zhou, J. T., & Szolovits, P. (2020). Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, the thirty-second innovative applications of artificial intelligence conference, IAAI 2020, the tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020* (pp. 8018–8025). AAAI Press. https://doi.org/10.1609/AAAI.V34I05.6311

Killamsetty, K., Zhao, X., Chen, F., & Iyer, R. (2021). RETRIEVE: Coreset selection for efficient and robust semi-supervised learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems* (pp. 14488–14501). Curran Associates, Inc. (*vol. 34*). https://proceedings.neurips.cc/paper_files/paper/2021/file/793bc52a941b3951dfdb85fb04f9fd06-Paper.pdf.

Kitaev, N., & Klein, D. (2018). Constituency parsing with a self-attentive encoder. In I. Gurevych, & Y. Miyao (Eds.), *Proceedings of the 56th annual meeting of the association for computational linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long papers* (pp. 2676–2686). Association for Computational Linguistics. https://aclanthology.org/P18-1249/. https://doi.org/10.18653/V1/P18-1249

Li, J., Ji, S., Du, T., Li, B., & Wang, T. (2019). TextBugger: Generating adversarial text against real-world applications. In *26th annual network and distributed system security symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society. https://www.ndss-symposium.org/ndss-paper/textbugger-generating-adversarial-text-against-real-world-applications/.

Li, L., Ma, R., Guo, Q., Xue, X., & Qiu, X. (2020). BERT-ATTACK: Adversarial attack against BERT using BERT. In B. Webber, T. Cohn, Y. He, & Y. Liu (Eds.), *Proceedings of the 2020 conference on empirical methods in natural language processing, EMNLP 2020, online, November 16-20, 2020* (pp. 6193–6202). Association for Computational Linguistics. https://doi.org/10.18653/V1/2020.EMNLP-MAIN.500

Li, M., Chen, L., Chen, J., He, S., Huang, H., Gu, J., & Zhou, T. (2023a). Reflection-tuning: Data recycling improves LLM instruction-tuning. arXiv preprint arXiv:2310.11716.

Li, M., Zhang, Y., He, S., Li, Z., Zhao, H., Wang, J., Cheng, N., & Zhou, T. (2024). Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. arXiv preprint arXiv:2402.00530.

Li, M., Zhang, Y., Li, Z., Chen, J., Chen, L., Cheng, N., Wang, J., Zhou, T., & Xiao, J. (2023b). From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. arXiv preprint arXiv:2308.12032.

Liang, J., Liao, L., Fei, H., Li, B., & Jiang, J. (2024). Actively learn from LLMs with uncertainty propagation for generalized category discovery. In *Proceedings of the 2024 conference of the North American chapter of the association for computational linguistics: Human language technologies (Volume 1: Long papers)* (pp. 7838–7851).

Lin, S., Hilton, J., & Evans, O. (2022). TruthfulQA: Measuring how models mimic human falsehoods. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (Volume 1: Long papers), ACL 2022, Dublin, Ireland, May 22-27, 2022* (pp. 3214–3252). Association for Computational Linguistics. https://doi.org/10.18653/V1/2022.ACL-LONG.229

Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., Zhou, D., Le, Q. V., Zoph, B., Wei, J. et al. (2023). The Flan collection: Designing data and methods for effective instruction tuning. In *International conference on machine learning* (pp. 22631–22648). PMLR.

Lu, Y., Bartolo, M., Moore, A., Riedel, S., & Stenetorp, P. (2022). Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (Volume 1: Long papers)* (pp. 8086–8098). Dublin, Ireland: Association for Computational Linguistics. https://aclanthology.org/2022.acl-long.556. https://doi.org/10.18653/v1/2022.acl-long.556

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605.

Maus, N., Chao, P., Wong, E., & Gardner, J. (2023). Black box adversarial prompting for foundation models. arXiv preprint arXiv:2302.04237.

Mindermann, S., Brauner, J. M., Razzak, M. T., Sharma, M., Kirsch, A., Xu, W., Höltgen, B., Gomez, A. N., Morisot, A., Farquhar, S. et al. (2022). Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International conference on machine learning* (pp. 15630–15649). PMLR.

Munteanu, A., Schwiegelshohn, C., Sohler, C., & Woodruff, D. (2018). On coresets for logistic regression. *Advances in Neural Information Processing Systems*, *31*, 6562–6571.

Naik, A., Ravichander, A., Sadeh, N. M., Rosé, C. P., & Neubig, G. (2018). Stress test evaluation for natural language inference. In E. M. Bender, L. Derczynski, & P. Isabelle

(Eds.), *Proceedings of the 27th International conference on computational linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018* (pp. 2340–2353). Association for Computational Linguistics. https://aclanthology.org/C18-1198/.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A. et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, *35*, 27730–27744.

Paul, M., Ganguli, S., & Dziugaite, G. K. (2021). Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, *34*, 20596–20607.

Peng, B., Li, C., He, P., Galley, M., & Gao, J. (2023). Instruction tuning with GPT-4. arXiv preprint arXiv:2304.03277.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. et al. (2018). Improving language understanding by generative pre-training, *OpenAI*, https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Ribeiro, M. T., Wu, T., Guestrin, C., & Singh, S. (2021). Beyond accuracy: Behavioral testing of NLP models with checklist (extended abstract). In Z. Zhou (Ed.), *Proceedings of the thirtieth international joint conference on artificial intelligence, IJCAI 2021, virtual event / Montreal, Canada, 19-27 August 2021* (pp. 4824–4828). ijcai.org. https://doi.org/10.24963/IJCAI.2021/659

Shu, M., Wang, J., Zhu, C., Geiping, J., Xiao, C., & Goldstein, T. (2023). On the exploitability of instruction tuning. *Advances in Neural Information Processing Systems*, *36*, 61836–61856.

Si, C., Gan, Z., Yang, Z., Wang, S., Wang, J., Boyd-Graber, J. L., & Wang, L. (2023). Prompting GPT-3 to be reliable. In *The eleventh international conference on learning representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/pdf?id=98p5x51L5af.

Sun, J., Shaib, C., & Wallace, B. C. (2023). Evaluating the zero-shot robustness of instruction-tuned language models. arXiv preprint arXiv:2306.11270.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., & Hashimoto, T. B. (2023). Stanford Alpaca: An instruction-following LLaMa model, *GitHub repository*, https://github.com/tatsu-lab/stanford_alpaca.

Toneva, M., Sordoni, A., Combes, R. T. d., Trischler, A., Bengio, Y., & Gordon, G. J. (2019). An empirical study of example forgetting during deep neural network learning. In *7th international conference on learning representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=BJlxm30cKm.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F. et al. (2023a). LLaMa: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S. et al. (2023b). LLaMa 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

Tsang, I. W., Kwok, J. T., Cheung, P.-M., & Cristianini, N. (2005). Core vector machines: Fast SVM training on very large data sets. *Journal of Machine Learning Research*, *6*, 363–392.

Wan, Z., Wang, Z., Wang, Y., Wang, Z., Zhu, H., & Satoh, S. (2024). Contributing dimension structure of deep feature for coreset selection. In M. J. Wooldridge, J. G. Dy, & S. Natarajan (Eds.), *Thirty-eighth AAAI conference on artificial intelligence, AAAI 2024, thirty-sixth conference on innovative applications of artificial intelligence, IAAI 2024, fourteenth symposium on educational advances in artificial intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada* (pp. 9080–9088). AAAI Press.https://doi.org/10.1609/AAAI.V38I8.28758

Wang, J., Hu, X., Hou, W., Chen, H., Zheng, R., Wang, Y., Yang, L., Ye, W., Huang, H., Geng, X., Jiao, B., Zhang, Y., & Xie, X. (2024). On the robustness of ChatGPT: An adversarial and out-of-distribution perspective. *IEEE Data Engineering Bulletin*, *47*(1), 48–62. http://sites.computer.org/debull/A24mar/p48.pdf.

Xia, X., Liu, J., Yu, J., Shen, X., Han, B., & Liu, T. (2023). Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *The eleventh international conference on learning representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/pdf?id=7D5EECbOaf9.

Xu, C., Sun, Q., Zheng, K., Geng, X., Zhao, P., Feng, J., Tao, C., & Jiang, D. (2023). WizardLM: Empowering large language models to follow complex instructions. arXiv preprint arXiv:2304.12244.

Xu, X., Zhang, J., Liu, F., Sugiyama, M., & Kankanhalli, M. S. (2023). Efficient adversarial contrastive learning via robustness-aware coreset selection. *Advances in Neural Information Processing Systems*, *36*, 75798–75825.

Yuan, B., Chen, Y., Zhang, Y., & Jiang, W. (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)* (pp. 10977–11011).

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., & Choi, Y. (2019). HellaSwag: Can a machine really finish your sentence? In A. Korhonen, D. R. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th conference of the association for computational linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long papers* (pp. 4791–4800). Association for Computational Linguistics. https://doi.org/10.18653/V1/P19-1472

Zhang, S., Dong, L., Li, X., Zhang, S., Sun, X., Wang, S., Li, J., Hu, R., Zhang, T., Wu, F. et al. (2023). Instruction tuning for large language models: A survey. arXiv preprint arXiv:2308.10792.

Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L. et al. (2023). LIMA: Less is more for alignment. *Advances in Neural Information Processing Systems*, *36*, 55006–55021.

Zhu, K., Wang, J., Zhou, J., Wang, Z., Chen, H., Wang, Y., Yang, L., Ye, W., Gong, N. Z., Zhang, Y. et al. (2023). PromptBench: Towards evaluating the robustness of large language models on adversarial prompts. arXiv preprint arXiv:2306.04528.