RICo: Refined In-Context Contribution for Automatic Instruction-Tuning Data Selection

Yixin Yang[1], Qingxiu Dong[1], Linli Yao[1], Fangwei Zhu[1], Weilin Luo[2], Bin Wang[2], Zhifang Sui[1]*

[1]State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University
[2]Huawei Noah's Ark Lab, China
yangyx@stu.pku.edu.cn, szf@pku.edu.cn

Abstract

Data selection for instruction tuning is crucial for improving the performance of large language models (LLMs) while reducing training costs. In this paper, we propose **R**efined **Co**ntribution Measurement with **I**n-**Co**ntext Learning (RICo), a novel gradient-free method that quantifies the fine-grained contribution of individual samples to both task-level and global-level model performance. RICo enables more accurate identification of high-contribution data, leading to better instruction tuning. We also introduce a lightweight selection paradigm trained on RICo scores, enabling scalable data selection with strictly linear inference complexity. Extensive experiments on 3 LLMs across 12 benchmarks and 5 pairwise evaluation sets demonstrate the effectiveness of RICo. Remarkably, on LLaMA3.1-8B, models trained in 15% of RICo-selected data outperform full datasets by 5.42 percentage points and exceed the best performance of widely used selection methods by 1.48 percentage points. We further analyze high-contribution samples selected by RICo, which show both diverse tasks and appropriate difficulty levels, rather than merely the most difficult cases.

Code — https://annayang2020.github.io/RICo_Data_Selection/

Extended version — https://arxiv.org/abs/2505.05327

Introduction

Recent advancements in large language models (LLMs), such as GPT-4o (Hurst et al. 2024), have demonstrated strong capabilities in both understanding and generation. These models are capable of performing a wide range of tasks (Team et al. 2024; Guo et al. 2025), often showing impressive flexibility and adaptability. Instruction tuning (Liu et al. 2023a; Xu et al. 2024b) has emerged as a powerful approach to enhance the performance of such models by finetuning them to better follow human instructions. While traditional instruction tuning relies on amassing vast datasets (Köpf et al. 2023; Chung et al. 2024), recent work (Zhou et al. 2023a) shows that manual selection of high-quality data subsets can achieve better performance with lower computational cost.

Automated data selection is crucial, as manual selection is costly and impractical. Previous studies on automated data selection (Xia et al. 2024; Wang et al. 2025) have explored gradient-based approaches to estimate sample value, but these are often computationally expensive. Others propose lightweight methods based on manually designed heuristics, such as textual features (Liu et al. 2023b; Bukharin and Zhao 2023; Zhang, Dai, and Peng 2025) and instruction difficulty (Li et al. 2023a, 2024), but these do not directly assess training impact and can introduce human inductive bias into the selection process. Recent works (Li et al. 2023c; Jiao et al. 2025) adopt binary comparisons in in-context learning (ICL) settings. While the implicit fine-tuning nature of ICL (Dong et al. 2022; Dai et al. 2023) shows promise in addressing prior limitations, existing approaches remain limited in robustly assessing sample value. They provide only coarse-grained signals, are often biased toward longer sample (Fang et al. 2024), and typically require a large number of inference calls.

In this paper, we introduce **R**efined **Co**ntribution Measurement with **I**n-**Co**ntext Learning (RICo), A novel gradient-free selection method that captures fine-grained sample contributions across both task-level and overall model performance. RICo measures each sample's contribution across diverse tasks while reducing biases such as length sensitivity and human inductive bias. We argue that such refined measurement enables more accurate identification of high-contribution data, leading to stronger instruction tuning results. Moreover, we train a lightweight selection paradigm on these contributions, reducing inference calls and enabling scalable data selection over large candidate pools.

Our approach measures the refined contribution of each sample through three components: 1) an assessment set that provides a reliable reference for model performance, 2) RICo scores that quantify the fine-grained contribution, and 3) a lightweight selection paradigm for scalable data selection. The RICo score reflects the contribution of each sample to task-specific and overall performance, including a fairness adjustment to mitigate length-related bias. A higher score indicates greater contribution, while a lower score reflects a more detrimental effect. Finally, the RICo-guided selection paradigm is efficiently trained with LoRA on the target model, reducing selection complexity to linear inference calls with respect to the number of training samples.

We demonstrate the effectiveness of RICo across multiple models and evaluation settings. Experiments on LLaMA3.1-8B (Grattafiori et al. 2024), Qwen2.5-3B (Yang et al. 2024),

*Corresponding author

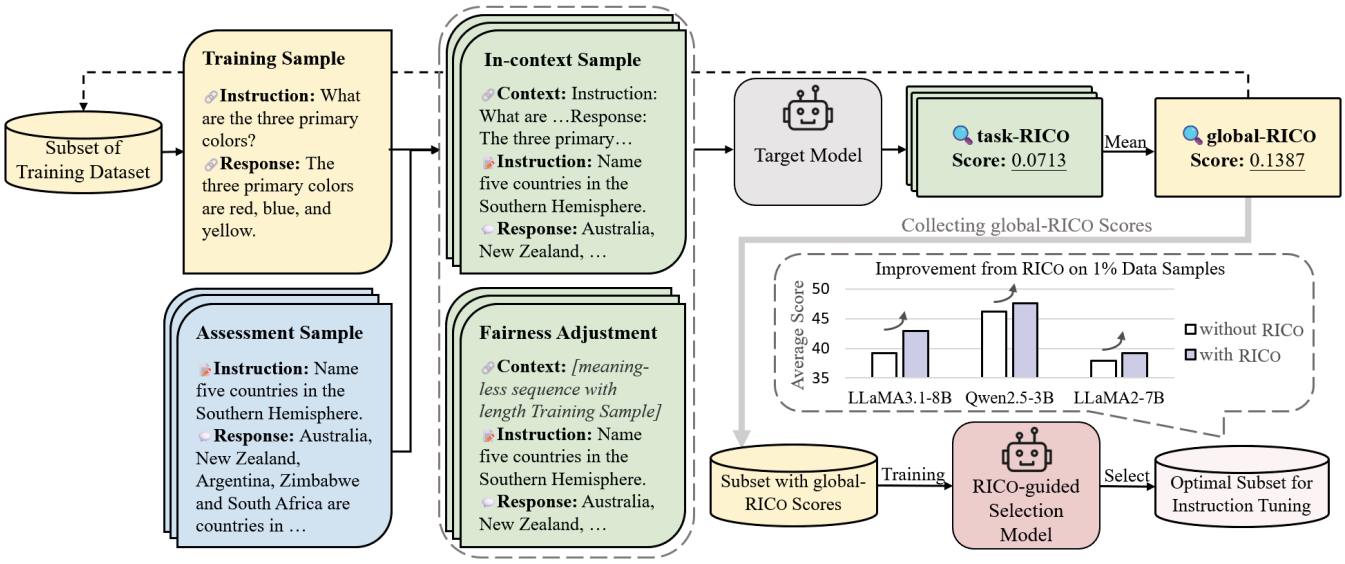

Figure 1: Overview of the RICo Method. The diagram illustrates the two key components of the method, including refined contribution quantification with the RICo score and RICo-guided selection paradigm training. This approach enables gradient-free, bias-reduced data selection with scalable inference.

and LLaMA2-7B (Lyu et al. 2024) are conducted using 12 widely used benchmarks and 5 pairwise comparison test sets. Models trained on a small fraction of RICo-selected data outperform their full-data counterparts. Notably, with only 15% of the data, the LLaMA3.1-8B and Qwen2.5-3B models improve by 5.42 and 1.24 percentage points, respectively; LLaMA2-7B achieves a 0.65-point gain using just 5%. RICo also surpasses widely used selection methods. For example, RICo improves LLaMA3.1-8B's average benchmark score by 1.48 points over the best prior method. Furthermore, we also analyze the impact of data scale, cross-dataset generalization, components ablation study and the properties of high-contribution samples selected by RICo. The main contributions of our paper are as follows:

- We propose RICo, a novel gradient-free data selection method that quantifies the refined contribution of individual samples to overall model performance, with fairness adjustments to reduce length-related bias.
- We introduce a lightweight selection paradigm trained on RICo contribution scores, enabling scalable data selection with a strictly linear inference complexity.
- We demonstrate the effectiveness of RICo through experiments on multiple models, 12 widely used benchmarks, and 5 pairwise test sets, showing consistent performance gains over baselines.

Related Work

Automatic Data Selection. Automatic data selection seeks to algorithmically construct optimal datasets to improve model performance, reduce training cost, mitigate undesirable model behaviors, and ensure evaluation quality (Albalak et al. 2024). With the rise of LLMs, it plays a central role across various training stages, including pretraining, instruction tuning, alignment, in-context learning, and task-specific finetuning. Pretraining selection focuses on filtering large-scale raw data (Soldaini et al. 2024); instruction tuning will be discussed in the next section; alignment involves model-based evaluation (Wang et al. 2023b) and reward model reweighting (Touvron et al. 2023); in-context learning emphasizes demonstration choice (Xu and Zhang 2024) and order (Lu et al. 2021); and task-specific tuning leverages utility-based (Ivison et al. 2023) or empirical methods (Grangier and Iter 2022).

Instruction-Tuning Data Selection. Automatic data selection for instruction tuning seeks data subsets that best enhance model performance. Gradient-based methods (Xia et al. 2024; Zhang et al. 2024; Wang et al. 2025) assess sample influence through gradients or subset training outcomes, but incur substantial computational cost. Heuristic-based methods (Liu et al. 2023b, 2024; Li et al. 2024; Zhang, Dai, and Peng 2025) use manually crafted indicators, which may not fully capture the true training impact and can introduce human inductive bias. See the extended version for all methods. In-context probing (ICP) methods (Li et al. 2023c; Jiao et al. 2025) leverage the implicit fine-tuning effect of ICL without model updates, reducing both human bias and computational cost. However, existing approaches typically rely on simple binary comparisons, yielding coarse-grained signals, exhibiting length bias, and incurring inference costs that grow multiplicatively with the size of the training pool. Inspired by the advantages of ICL, RICo introduces a fine-grained, bias-reduced contribution measurement and a lightweight selection paradigm that enables scalable data selection with strictly linear inference complexity.

Methodology

To efficiently identify influential samples, we propose RICo, a gradient-free framework for refined contribution estimation. RICo builds on ICL's implicit fine-tuning nature, leveraging the model's attention mechanism to simulate parameter updates through demonstration processing (Dai et al. 2023). Based on this mechanism, RICo further captures fine-grained sample influence on instruction tuning. As illustrated in Figure 1, RICo consists of three components: 1) constructing a representative assessment set, 2) computing the RICo score to quantify fine-grained sample contribution, and 3) training a RICo-guided selection model to curate the full dataset with strictly linear inference complexity.

Assessment Set Construction

The goal of this component is to construct an assessment set that spans diverse tasks and reliably reflects the model's overall capabilities. To promote reliability and mitigate bias in data origin and task design, we sample from three sources: ChatGPT-generated, GPT-4-generated, and human-authored data, resulting in a high-quality set covering various task types and instruction complexities. The assessment set is used solely for assessment, with no overlap with training data or downstream benchmarks, ensuring independence and fair comparison. Formally, we define the assessment set as D_a, consisting of n instruction tuning samples, each represented as a pair (x^a, y^a), where $x^a = \text{map}(\text{Instruction}, [\text{Input}])$ is the full instruction formed by concatenating the instruction and the optional input, and y^a is the corresponding response. Meanwhile, the candidate training set is denoted as $D_t = \{(x_1^t, y_1^t), (x_2^t, y_2^t), \ldots, (x_m^t, y_m^t)\}$, where m is the number of instruction-response pairs.

Refined Contribution Quantification

We introduce the RICo score, a principled metric for quantifying the refined contribution of individual samples to model performance. It enables low-cost, bias-reduced estimation of task-level contributions, culminating in a fine-grained overall contribution.

Perplexity. Perplexity directly evaluates the likelihood of response tokens and provides a smoother measure of model performance compared to accuracy. It aligns well with instruction tuning objectives, making it a commonly adopted metric for both evaluation and data selection in instruction-tuning tasks (Li et al. 2023a,c, 2024). Specifically, for a task sample $S_i = (x_i^a, y_i^a) \in D_a$, where D_a is the assessment set, y_i^a consists of N tokens, and the k-th token is denoted as $y_{i,k}^a$. The model's performance under parameters θ on task i is defined as shown in Equation 1. To evaluate the contribution of a training sample $T_j = (x_j^t, y_j^t)$ from a subset $D_t' \subseteq D_t$, the sample is inserted into the context during inference. The model's performance on S_i is then reassessed after conditioning on T_j. The perplexity in this setting is denoted as Equation 2.

$$\begin{aligned}\text{PPL}_\theta(S_i) &\triangleq \text{PPL}_\theta(y_i^a | x_i^a) \\ &= \exp\left(-\frac{1}{N}\sum_{k=1}^{N} \log p_\theta(y_{i,k}^a | x_i^a, y_{i,1}^a, ..., y_{i,k-1}^a)\right)\end{aligned} \tag{1}$$

$$\text{PPL}_\theta(S_i \mid T_j) \triangleq \text{PPL}_\theta(y_i^a \mid T_j, x_i^a) \tag{2}$$

Fairness Adjustment. Despite its wide use, we observe that perplexity tends to decrease with longer inputs, even when the content is uninformative. This length sensitivity introduces a bias toward longer samples and fails to fairly reflect true sample contribution. To address this, we introduce a fairness adjustment to neutralize length effects. Specifically, we define the baseline performance on S_i by replacing T_j with a randomly generated, semantically meaningless sequence T_j^{rand} of the same length. The length-controlled reference perplexity is defined as:

$$\text{PPL}_\theta(S_i \mid T_j^{\text{rand}}) \triangleq \text{PPL}_\theta(y_i^a \mid T_j^{\text{rand}}, x_i^a) \tag{3}$$

Task-level RICo Score. We further define the task-level RICo score to quantify the refined contribution of an individual sample on model behavior for a specific task. The score integrates the above fairness adjustment to reduce length bias. We normalize task difficulty to reduce its impact on contribution scoring, so that the score primarily reflects the sample's effect rather than task difficulty. Specifically, we define the task-level RICo score of sample T_j on task i under parameters θ as shown in Equation 4, where ϵ is a small constant to avoid division by zero. A higher RICo score implies greater contribution of T_j to the model's performance on task i.

$$\text{task-RICo}_\theta(T_j \to S_i) \triangleq \frac{\text{PPL}_\theta(S_i|T_j^{\text{rand}}) - \text{PPL}_\theta(S_i|T_j)}{\text{PPL}_\theta(S_i) + \epsilon} \tag{4}$$

$$\text{global-RICo}_\theta(T_j) \triangleq \frac{1}{n}\sum_{S_i \in D_a} \text{task-RICo}_\theta(T_j \to S_i) \tag{5}$$

Global-level RICo Score. To further quantify a sample's overall contribution to model performance, we compute the global-level RICo score by aggregating its task-level scores. The diversity and coverage of the assessment set provide a reliable reference for overall model performance. To promote generalization rather than specialization, we assign equal weight to all tasks. The global-level score, shown in Equation 5, reflects the refined overall contribution of a sample and enables ranking based on total impact. Higher global-level RICo scores indicate greater overall contribution.

RICo-guided Selection Paradigm

To further improve selection efficiency, we train an RICo-guided selection model using the computed global-level RICo scores. This reduces the selection complexity from $O(nm)$ to $O(m)$ inference calls, where n is the number of assessment samples and m is the number of training candidates. As a result, it enables efficient and scalable identification of high-contribution samples for instruction tuning. Specifically, we sample a subset of training data and label the top $K\%$ of samples by global-level RICo score as high-contribution.

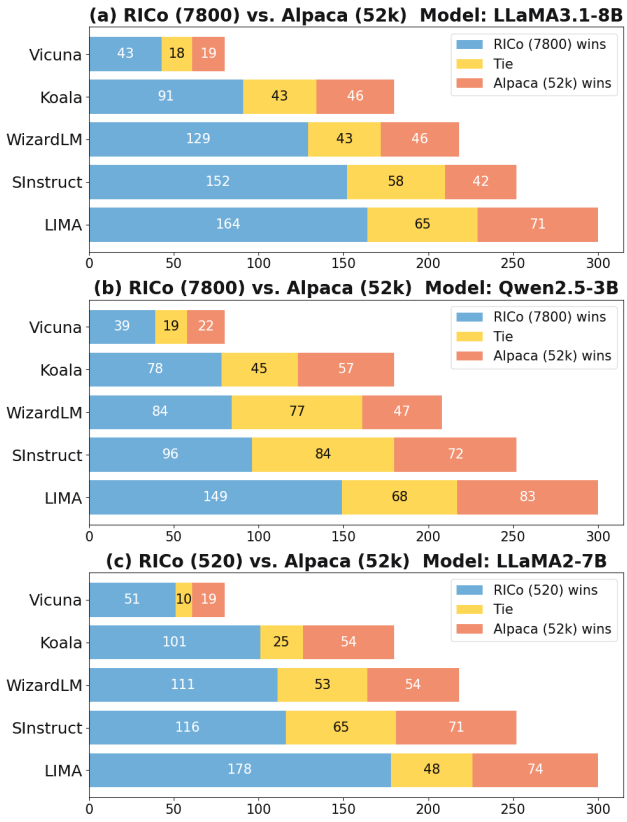

Figure 2: Pairwise win-tie-lose comparison between models trained on RICo-selected subsets and those trained on the full Alpaca dataset across five evaluation benchmarks. RICo-selected data enhances instruction-following ability with fewer samples, as evidenced by pairwise comparisons.

A selection model is then trained using LoRA on the target model to classify high-contribution samples. Once trained, this selection paradigm can be applied to the full candidate pool, enabling linear-time selection in large-scale settings, with potential for practical use in realistic applications.

Experiments Setup

Datasets

Training Dataset. In this paper, we use the classic Alpaca dataset (Taori et al. 2023), one of the original instruction datasets created by Stanford University. The Alpaca dataset consists of 52,002 instruction-following samples, generated with TextDavinci-003 through the self-instruction approach (Wang et al. 2022). The WizardLM dataset (Xu et al. 2024a) utilizes the Evol-Instruct algorithm to enhance the quality and complexity of instruction data. We conduct experiments using WizardLM-70K to verify the generalization of our method across different datasets.

Assessment Dataset. We measure the impact of a training candidate sample on model performance over the assessment dataset, as described in previous section. The assessment set is used solely for evaluation, with no overlap with training data or downstream benchmarks. To ensure diversity and reduce bias, we randomly select 1,020 instructions from three sources: OpenOrca-GPT3.5 (Lian et al. 2023) (ChatGPT-generated data), OpenOrca-GPT4 (GPT-4-generated data), and Dolly-15K (Mike et al. 2023) (human-generated data).

Implementation Details

For experiments on LLaMA3.1-8B (Grattafiori et al. 2024), Qwen2.5-3B (Yang et al. 2024), and LLaMA2-7B (Lyu et al. 2024), we follow the original Alpaca[1] training configuration and codebase. Models are trained for 3 epochs using Adam with a learning rate of 2×10^{-5} and batch size 128. The maximum input length is 512 for Alpaca and 2048 for WizardLM. The selection paradigm is trained on 4,500 samples with LoRA for 5 epochs, using a learning rate of 5×10^{-5}, rank 8, and $\alpha = 8$. Experiments run on Ubuntu 20.04 with A40 GPUs, 502 GiB RAM, Python 3.9.19, PyTorch 2.4.0, and CUDA 12.1.

Evaluation Metrics

Benchmark Evaluation. To evaluate the effectiveness of RICo, we report performance on several widely used benchmarks. These benchmarks fall into three categories, assessing different capabilities: instruction following, knowledge, and reasoning. Instruction following evaluation includes: IFEval (Zhou et al. 2023c) and AlpacaEval (Li et al. 2023b). AlpacaEval provides automatic evaluation on the AlpacaFarm set, where we use GPT-4 as both the response generator and evaluator. Knowledge evaluation includes: GLUE (Wang et al. 2018), GPQA (Rein et al. 2024), MMLU (Hendrycks et al. 2020), and TruthfulQA (Lin, Hilton, and Evans 2021). Reasoning evaluation includes: ARC (Clark et al. 2018), BBH (Suzgun et al. 2022), HellaSwag (Zellers et al. 2019), LogiQA (Liu et al. 2020), MuSR (Sprague et al. 2023), and Winogrande (Sakaguchi et al. 2021).

Pairwise Comparison. We also evaluate instruction-following performance on five pairwise comparison sets: WizardLM (Xu et al. 2024a), Self-Instruct (SInstruct) (Wang et al. 2022), Vicuna(Chiang et al. 2023), Koala (Vu et al. 2023), and LIMA (Zhou et al. 2023b), containing 218, 252, 80, and 300 human-curated instructions across domains like math, coding, writing, and general knowledge. GPT-4 serves as the judge, scoring model responses in pairwise comparisons from 1 to 10 based on relevance, accuracy, and other factors. To reduce positional bias (Wang et al. 2023a), each pair is submitted twice with reversed order. A model is considered to win only if it does not lose in both orders, following these rules: **Win**, wins both, or wins one and ties the other; **Tie**, ties both, or wins one and loses one; **Lose**, loses both, or loses one and ties the other.

Experiment Results

Main Results

In this section, we present the evaluation benchmark results shown in Table 1. The models are trained using different proportions of Alpaca data selected by the RICo method,

[1]https://github.com/tatsu-lab/stanford_alpaca

	Instruction Following↑		Knowledge↑				Reasoning↑						Avg↑	FLOPs↓
	IFEval	AlpacaEval	GLUE	GPQA	MMLU	TQA	ARC	BBH	HS	LQA	MuSR	WG		($\times 10^{12}$)
LLaMA3.1-8B														
FULL	15.68	18.66	54.48	27.27	43.26	39.60	40.53	36.46	51.42	26.73	37.51	63.77	37.95	40.12
RICo (1%)	8.31	28.62	58.47	31.31	55.77	43.11	49.91	44.51	61.19	23.35	39.37	71.03	42.91(+4.96)	**0.59**
RICo (5%)	16.21	28.96	56.91	27.27	53.70	40.51	47.18	44.23	58.44	24.42	37.75	69.46	42.09(+4.14)	3.06
RICo (10%)	16.08	27.99	57.86	26.76	53.57	42.63	45.48	42.81	56.27	26.57	42.01	66.93	42.08(+4.13)	5.62
RICo (15%)	21.16	29.89	60.58	30.81	52.30	43.16	44.80	44.15	55.56	28.42	41.64	67.96	**43.37(+5.42)**	8.37
Qwen2.5-3B														
FULL	32.34	28.74	68.98	31.82	61.71	45.42	46.42	45.05	55.13	35.79	44.51	63.85	46.65	36.46
RICo (1%)	27.04	34.55	67.22	31.82	64.63	44.08	53.16	47.35	56.19	34.71	41.05	68.51	47.53(+0.88)	**0.44**
RICo (5%)	20.85	36.25	68.34	27.88	64.73	43.26	53.50	47.80	58.10	32.10	39.72	69.06	46.80(+0.15)	2.27
RICo (10%)	25.28	37.05	69.26	34.85	63.67	43.35	50.00	47.00	57.85	35.79	40.25	68.27	47.72(+1.07)	5.28
RICo (15%)	23.07	30.11	70.46	34.34	64.09	46.90	51.71	47.68	58.16	38.40	41.96	67.80	**47.89(+1.24)**	6.85
LLaMA2-7B														
FULL	15.48	21.76	53.98	26.26	43.69	40.48	41.72	36.18	53.98	25.03	41.11	62.82	38.54	48.72
RICo (1%)	5.25	30.35	54.51	23.23	40.55	45.82	42.24	38.34	59.20	25.19	37.68	67.96	**39.19(+0.65)**	**0.66**
RICo (5%)	7.78	31.47	51.93	23.23	36.95	45.72	40.53	36.07	57.57	23.96	40.62	66.77	38.55(+0.01)	3.26
RICo (10%)	11.86	29.76	50.62	25.76	37.30	45.19	41.72	35.38	57.39	24.88	39.82	66.38	38.84(+0.30)	6.16
RICo (15%)	8.06	26.83	54.67	27.78	36.39	43.89	42.66	34.74	57.01	23.50	39.80	66.06	38.45(-0.09)	9.18

Table 1: Performance of three models in terms of FLOPs and evaluation benchmarks for instruction following, knowledge, and reasoning. The datasets TQA, HS, LQA, and WG correspond to TruthfulQA, HellaSwag, LogiQA, and WinoGrande, respectively. Models trained on RICo-selected data outperform full-dataset training, achieving better results with fewer samples.

including 1%, 5%, 10%, and 15%, corresponding to 520, 2,600, 5,200, and 7,800 samples, respectively. FULL denotes models trained on the full, unfiltered dataset. Experiments are conducted on multiple models, including LLaMA3.1-8B, Qwen2.5-3B and LLaMA2-7B. Models trained on a small fraction of RICo-selected data often match or exceed the performance of full-data models across all benchmarks, with consistent gains in average benchmark scores. Specifically, the LLaMA3.1-8B model trained on just 15% of RICo-selected data exceeds the full-data baseline by 5.42 percentage points. Similarly, the Qwen2.5-3B model with 15% RICo-selected data and the LLaMA2-7B model with only 1% achieve overall gains of 1.24% and 0.65%, respectively. Notably, these RICo-trained models require only about 1/5 to 1/100 of the FLOPs compared to full-data training, demonstrating significantly lower resource consumption. These results underscore the effectiveness of RICo in enhancing instruction tuning with substantially less data.

The results of the pairwise evaluation, including detailed win-tie-lose statistics on the Vicuna, Koala, WizardLM, SInstruct, and LIMA test sets, are presented in Figure 2. We focus on the best-performing models from the evaluation benchmarks: LLaMA3.1-8B, Qwen2.5-3B, and LLaMA2-7B, each trained on 15%, 15%, and 1% RICo-selected samples, respectively. All three models consistently outperform their counterparts trained on the full Alpaca dataset across all five test sets. These results further validate the effectiveness of RICo in improving instruction-following ability, even with significantly less training data.

	IF	KN	RS	Avg
Random	12.13	45.18	46.89	40.52
Low PPL	19.35	44.45	46.53	41.31
Top PPL	3.28	41.72	44.69	36.80
Alpagasus	16.39	45.14	45.29	40.42
Deita	15.20	45.46	45.92	40.65
Superfilter	20.25	44.76	45.45	41.02
Nuggets	18.63	44.56	46.22	41.07
LESS	23.56	45.55	45.56	41.89
ArmoRM	16.79	40.77	46.53	39.65
SelectIT	13.42	44.25	45.74	39.86
RICo (Ours)	**25.52**	**46.71**	**47.09**	**43.37**

Table 2: Performance of different methods on evaluation benchmarks. The 'IF', 'KN', and 'RS' correspond to average scores on Instruction Following, Knowledge, and Reasoning, respectively. The models are trained on LLaMA3.1-8B with 15% selected samples. RICo outperforms widely used data selection methods on evaluation benchmarks.

Comparison with Other Methods

In this section, we compare our method with several widely used data selection baselines on the Alpaca dataset, as shown in Table 2 and Table 3. *Random* selects samples uniformly at random. *Low-PPL* and *Top-PPL* select samples with the lowest and highest perplexity scores, respectively. *Alpagasus* (Chen et al. 2023) uses GPT-3.5-Turbo to score responses based on helpfulness, accuracy, and other dimensions. *Deita* (Liu et al. 2023b) leverages models fine-tuned from

Comparison Test Set	Pairwise Winning Score ↑ Vicuna	WizardLM	LIMA	SInstruct	Koala	Overall
RICo vs. Random	1.2125	1.4541	1.3233	1.5159	1.3389	1.3922
RICo vs. Low PPL	1.3375	1.3945	1.4533	1.3690	1.3333	1.3903
RICo vs. Top PPL	1.9125	1.8073	1.8567	1.8452	1.7778	1.8340
RICo vs. Alpagasus (Chen et al. 2023)	1.1250	1.1789	1.1033	1.3611	1.3389	1.2252
RICo vs. Deita (Liu et al. 2023b)	1.3375	1.2661	1.2733	1.3730	1.3889	1.3214
RICo vs. Superfilter (Li et al. 2024)	1.0375	1.0780	1.0100	1.1627	1.0944	1.0786
RICo vs. Nuggets (Li et al. 2023c)	1.1625	1.1422	1.0967	1.1508	1.1111	1.1272
RICo vs. LESS (Xia et al. 2024)	1.0125	1.1101	0.9467	1.0079	0.9500	1.0019
RICo vs. ArmoRM (Wang et al. 2024)	1.1750	1.2982	1.0967	1.4229	1.1611	1.2367
RICo vs. SelectIT (Liu et al. 2024)	1.3375	1.3303	1.3267	1.4008	1.2611	1.3350

Table 3: Comparison with other methods on Vicuna, WizardLM, LIMA, SInstruct, and Koala test set. The pairwise Winning Scores are calculated between models using our method and other methods. All the comparisons are performed by GPT-4, and the values that are greater than 1.0 represent our models are better and vice versa. The models are trained on LLaMA3.1-8B with 15% selected samples. RICo consistently outperforms widely used data selection methods on pairwise comparison test sets.

LLaMA and Mistral to assess sample quality, complexity, and diversity. *Superfilter* (Li et al. 2024) ranks samples using IFD scores derived from GPT-2 to estimate instruction difficulty, assuming that smaller models provide sufficiently reliable signals. *Nuggets* (Li et al. 2023c), a representative ICP-based method, assigns golden scores via binary comparisons in the ICL setting. We use its officially implemented KMeans_{100} selection method. *LESS* (Xia et al. 2024) is a representative gradient-based method, which selects the most influential training points by modeling the impact between candidate and validation samples. *ArmoRM* (Wang et al. 2024) is the state-of-the-art open-sourced reward model. *SelectIT* (Liu et al. 2024) quantifies model uncertainty via token-, sentence-, and model-level reflection.

Table 2 presents results on 13 evaluation benchmarks, reporting average scores in instruction tuning, knowledge, reasoning, and overall performance. RICo achieves the best performance across all categories, demonstrating its effectiveness in selecting high-contribution data for instruction tuning. Detailed results are provided in the extended version. Table 3 presents the pairwise winning scores of the RICo model against other data selection baselines. Models are trained on 15% selected data using LLaMA3.1-8B. The score is computed as $(\text{Num(Win)} - \text{Num(Lose)})/\text{Num(All)} + 1$, directly comparing RICo against each baseline. Scores above 1 indicate RICo performs better, with higher values showing greater advantage. RICo consistently achieves superior overall performance compared to all baselines.

Analysis

Optimal Data Scale for Instruction Tuning. We examine how data scale impacts performance using LLaMA3.1-8B trained on RICo-selected Alpaca subsets ranging from 1% to 100%. As shown in Figure 3, the average score across all evaluation benchmarks generally rises and then declines, peaking at 15%, suggesting an optimal scale for instruction tuning. The performance drop beyond this point suggests that low-value samples dilute the training signal, highlighting RICo's ability to prioritize valuable data. Based on the observation, we choose the model trained on 15% RICo-selected data as the representative for comparisons. Detailed results are provided in the extended version.

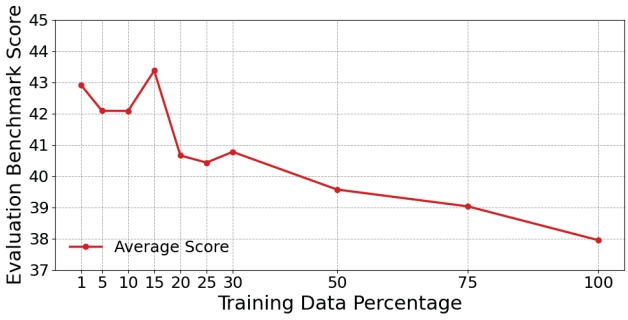

Figure 3: Model performance (average score) on evaluation benchmarks with varying proportions of RICo-selected Alpaca data. The models are trained based on LLaMA3.1-8B. The dataset is Alpaca. As the data scale increases, performance first improves and then declines, peaking at 15%.

Generalization of RICo on different Datasets. Beyond Alpaca, we evaluate RICo's generalization by applying the selection paradigm trained on Alpaca to the WizardLM dataset. As shown in Table 4 and detailed in the extended version, we test selection scales of 1%, 5%, 10%, 15%, and 100%, corresponding to 700, 3,500, 7,000, 10,500, and 70,000 samples. Models trained on RICo-selected subsets consistently outperform those trained on the full dataset across across all categories, achieving higher average scores. Notably, the model trained on just 5% of the data achieves the best results, with a 4.54 percentage point improvement in average evaluation benchmark score. These results demonstrate that both RICo and its selection paradigm generalize effectively across datasets.

Ablation Study on RICo Components. To analyze the impact of each RICo component, we perform an ablation study using 1% of training data with LLaMA3.1-8B. This setting is computationally efficient while still sufficient to reveal performance differences. We compare the full RICo

	IF	KN	RS	Avg
FULL	33.61	45.46	44.37	42.94
RICo (1%)	34.57	46.39	**49.57**	46.01
RICo (5%)	**43.21**	**48.28**	48.36	**47.48**
RICo (10%)	36.89	48.15	47.18	45.79
RICo (15%)	35.24	46.29	45.76	44.18

Table 4: Evaluation benchmark results of models trained on WizardLM dataset and its RICo-selected subsets. The 'IF', 'KN', and 'RS' correspond to average scores on Instruction Following, Knowledge, and Reasoning, respectively. Results on WizardLM dataset confirm the generalization of RICo across datasets.

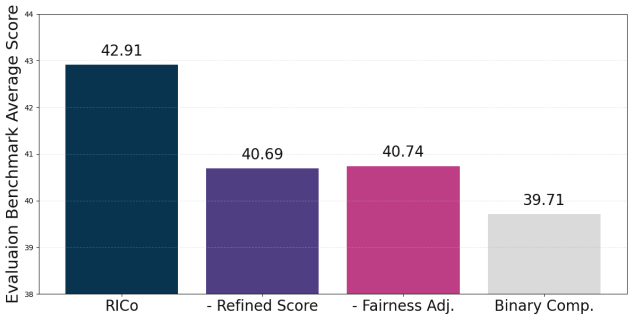

Figure 4: Ablation study on the contribution of components in RICo. '- Refined Score' removes continuous scoring. '- Fairness Adj.' removes the fairness adjustment. 'Binary Comp.' removes both components. Results show that both components individually improve performance, and their combination achieves the best overall result.

method with three variants: (1) *RICo - Refined Score*, which replaces continuous scores with binary helpful/unhelpful labels; (2) *RICo - Fairness Adjustment*, which removes the length-controlled adjustment, leaving input length bias uncorrected; and (3) *Binary Comparison*, which removes both components, reducing RICo to basic binary comparisons. As shown in Figure 4, each component individually improves performance. Refined scores capture subtle differences in sample contribution, while fairness adjustment mitigates length bias. Their combination achieves the best results, indicating that both components are complementary and jointly essential for accurate contribution estimation.

Characteristics of High-Contribution RICo Samples. We further analyze the distribution of high-contribution samples selected by RICo. As shown in Figure 5, they exhibit both cross-cluster dispersion and local density in t-SNE space. This pattern aligns with the notion in previous studies (Liu et al. 2023b; Bukharin and Zhao 2023) that 'instruction data should be diverse'. However, the distribution is uneven, with denser regions suggesting that high-contribution samples tend to exhibit certain clustering behavior in feature space. Figure 6 compares instruction difficulty between high-contribution samples and the full dataset using the IFD score (Li et al. 2023a, 2024). A t-test (Kendall 1937) (t = 2.24, p = 0.0251) shows that high-contribution samples are significantly more difficult on average, suggesting that the selection method favors more challenging data. However, these samples are not exclusively in the highest difficulty range. Some mid-level samples are retained, while extremely difficult ones are avoided. This challenges the assumption that harder samples are always more beneficial for training (Li et al. 2023a, 2024; Zhang, Dai, and Peng 2025).

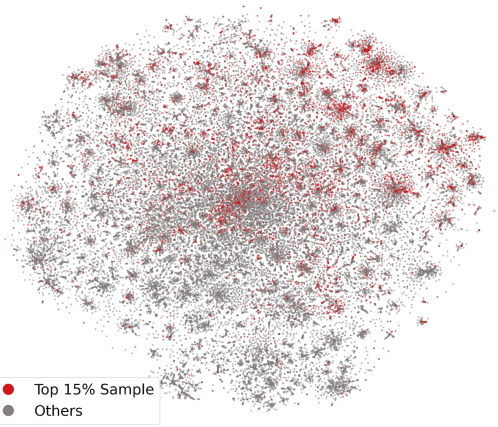

Figure 5: t-SNE visualization of Alpaca embeddings. Red denotes the top 15% RICo samples; gray denotes the rest. High-contribution samples are diverse.

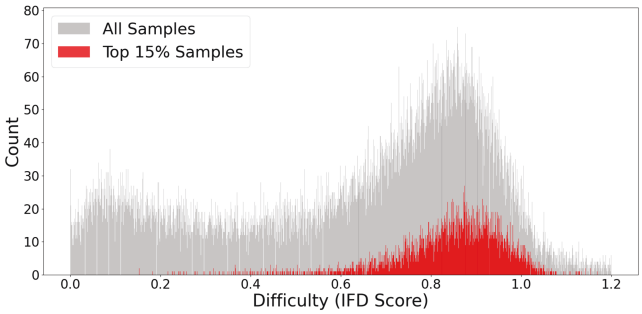

Figure 6: Difficulty distribution of top 15% high-contribution samples vs. full Alpaca dataset. High-contribution samples (red bars), selected via RICo for instruction tuning, exhibit distinct difficulty patterns compared to the full dataset (gray bars). High-contribution RICo samples tend to fall within an appropriate difficulty range.

Conclusion

We propose RICo, a gradient-free data selection method that quantifies individual sample contributions for instruction tuning. By computing task- and global-level scores with reduced bias, RICo identifies high-contribution data. We also introduce a lightweight paradigm for efficient data curation. Experiments on 3 LLMs across 12 benchmarks and 5 pairwise test sets show that RICo-selected data outperforms full-data and other methods, achieving better performance with fewer samples. We further analyze optimal data scales, generalization, ablation, and sample characteristics.

Acknowledgements

This paper is supported by NSFC project 62476009.

References

Albalak, A.; Elazar, Y.; Xie, S. M.; Longpre, S.; Lambert, N.; Wang, X.; Muennighoff, N.; Hou, B.; Pan, L.; Jeong, H.; et al. 2024. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*.

Bukharin, A.; and Zhao, T. 2023. Data Diversity Matters for Robust Instruction Tuning. *ArXiv*, abs/2311.14736.

Chen, L.; LI, S.; Yan, J.; Wang, H.; Gunaratna, K.; Yadav, V.; Tang, Z.; Srinivasan, V.; Zhou, T.; Huang, H.; and Jin, H. 2023. AlpaGasus: Training A Better Alpaca with Fewer Data. *ArXiv*, abs/2307.08701.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3): 6.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70): 1–53.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Dai, D.; Sun, Y.; Dong, L.; Hao, Y.; Ma, S.; Sui, Z.; and Wei, F. 2023. Why Can GPT Learn In-Context? Language Models Secretly Perform Gradient Descent as Meta-Optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, 4005–4019.

Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Ma, J.; Li, R.; Xia, H.; Xu, J.; Wu, Z.; Liu, T.; et al. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Fang, L.; Wang, Y.; Liu, Z.; Zhang, C.; Jegelka, S.; Gao, J.; Ding, B.; and Wang, Y. 2024. What is Wrong with Perplexity for Long-context Language Modeling? *ArXiv*, abs/2410.23771.

Grangier, D.; and Iter, D. 2022. The Trade-offs of Domain Adaptation for Neural Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3802–3813.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; Clark, A.; Ostrow, A.; Welihinda, A.; Hayes, A.; Radford, A.; et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Ivison, H.; Smith, N. A.; Hajishirzi, H.; and Dasigi, P. 2023. Data-Efficient Finetuning Using Cross-Task Nearest Neighbors. In *Findings of the Association for Computational Linguistics: ACL 2023*, 9036–9061.

Jiao, C.; Gao, W.; Raghunathan, A.; and Xiong, C. 2025. On the Feasibility of In-Context Probing for Data Attribution. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 5140–5155.

Kendall, M. G. 1937. Statistical Methods for Research Workers. *Nature*, 139: 737–737.

Köpf, A.; Kilcher, Y.; Von Rütte, D.; Anagnostidis, S.; Tam, Z. R.; Stevens, K.; Barhoum, A.; Nguyen, D.; Stanley, O.; Nagyfi, R.; et al. 2023. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36: 47669–47681.

Li, M.; Zhang, Y.; He, S.; Li, Z.; Zhao, H.; Wang, J.; Cheng, N.; and Zhou, T. 2024. Superfiltering: Weak-to-Strong Data Filtering for Fast Instruction-Tuning. In *Annual Meeting of the Association for Computational Linguistics*.

Li, M.; Zhang, Y.; Li, Z.; Chen, J.; Chen, L.; Cheng, N.; Wang, J.; Zhou, T.; and Xiao, J. 2023a. From Quantity to Quality: Boosting LLM Performance with Self-Guided Data Selection for Instruction Tuning. *ArXiv*, abs/2308.12032.

Li, X.; Zhang, T.; Dubois, Y.; Taori, R.; Gulrajani, I.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023b. AlpacaEval: An Automatic Evaluator of Instruction-following Models. https://github.com/tatsu-lab/alpaca_eval.

Li, Y.; Hui, B.; Xia, X.; Yang, J.; Yang, M.; Zhang, L.; Si, S.; Chen, L.-H.; Liu, J.; Liu, T.; et al. 2023c. One-shot learning as instruction data prospector for large language models. *arXiv preprint arXiv:2312.10302*.

Lian, W.; Goodson, B.; Pentland, E.; Cook, A.; Vong, C.; and "Teknium". 2023. OpenOrca: An open dataset of GPT augmented FLAN reasoning traces.

Lin, S.; Hilton, J.; and Evans, O. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Liu, F.; Wang, X.; Yao, W.; Chen, J.; Song, K.; Cho, S.; Yacoob, Y.; and Yu, D. 2023a. MMC: Advancing Multimodal Chart Understanding with Large-scale Instruction Tuning. *ArXiv*, abs/2311.10774.

Liu, J.; Cui, L.; Liu, H.; Huang, D.; Wang, Y.; and Zhang, Y. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*.

Liu, L.; Liu, X.; Wong, D. F.; Li, D.; Wang, Z.; Hu, B.; and Zhang, M. 2024. SelectIT: Selective Instruction Tuning for LLMs via Uncertainty-Aware Self-Reflection. In *Neural Information Processing Systems*.

Liu, W.; Zeng, W.; He, K.; Jiang, Y.; and He, J. 2023b. What Makes Good Data for Alignment? A Comprehensive Study of Automatic Data Selection in Instruction Tuning. *ArXiv*, abs/2312.15685.

Lu, Y.; Bartolo, M.; Moore, A.; Riedel, S.; and Stenetorp, P. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Lyu, K.; Zhao, H.; Gu, X.; Yu, D.; Goyal, A.; and Arora, S. 2024. Keeping llms aligned after fine-tuning: The crucial role of prompt templates. *arXiv preprint arXiv:2402.18540*.

Mike, C.; Matt, H.; Ankit, M.; Jianwei, X.; Jun, W.; Sam, S.; Ali, G.; Patrick, W.; Matei, Z.; and Reynold, X. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Rein, D.; Hou, B. L.; Stickland, A. C.; Petty, J.; Pang, R. Y.; Dirani, J.; Michael, J.; and Bowman, S. R. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Sakaguchi, K.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9): 99–106.

Soldaini, L.; Kinney, R.; Bhagia, A.; Schwenk, D.; Atkinson, D.; Authur, R.; Bogin, B.; Chandu, K.; Dumas, J.; Elazar, Y.; et al. 2024. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*.

Sprague, Z.; Ye, X.; Bostrom, K.; Chaudhuri, S.; and Durrett, G. 2023. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. *arXiv preprint arXiv:2310.16049*.

Suzgun, M.; Scales, N.; Schärli, N.; Gehrmann, S.; Tay, Y.; Chung, H. W.; Chowdhery, A.; Le, Q. V.; Chi, E. H.; Zhou, D.; et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Stanford alpaca: an instruction-following llama model (2023).

Team, G.; Mesnard, T.; Hardin, C.; Dadashi, R.; Bhupatiraju, S.; Pathak, S.; Sifre, L.; Rivière, M.; Kale, M. S.; Love, J.; et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

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

Vu, T.-T.; He, X.; Haffari, G.; and Shareghi, E. 2023. Koala: An index for quantifying overlaps with pre-training corpora. *arXiv preprint arXiv:2303.14770*.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Wang, H.; Xiong, W.; Xie, T.; Zhao, H.; and Zhang, T. 2024. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*.

Wang, J.; Lin, X.; Qiao, R.; Koh, P. W.; Foo, C.-S.; and Low, B. K. H. 2025. NICE Data Selection for Instruction Tuning in LLMs with Non-differentiable Evaluation Metric. In *Forty-second International Conference on Machine Learning*.

Wang, P.; Li, L.; Chen, L.; Cai, Z.; Zhu, D.; Lin, B.; Cao, Y.; Liu, Q.; Liu, T.; and Sui, Z. 2023a. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

Wang, T.; Yu, P.; Tan, X. E.; O'Brien, S.; Pasunuru, R.; Dwivedi-Yu, J.; Golovneva, O.; Zettlemoyer, L.; Fazel-Zarandi, M.; and Celikyilmaz, A. 2023b. Shepherd: A critic for language model generation. *arXiv preprint arXiv:2308.04592*.

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Xia, M.; Malladi, S.; Gururangan, S.; Arora, S.; and Chen, D. 2024. LESS: Selecting Influential Data for Targeted Instruction Tuning. *ArXiv*, abs/2402.04333.

Xu, C.; Sun, Q.; Zheng, K.; Geng, X.; Zhao, P.; Feng, J.; Tao, C.; Lin, Q.; and Jiang, D. 2024a. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Xu, S.; and Zhang, C. 2024. Misconfidence-based demonstration selection for llm in-context learning. *arXiv preprint arXiv:2401.06301*.

Xu, Z.; Jiang, F.; Niu, L.; Deng, Y.; Poovendran, R.; Choi, Y.; and Lin, B. Y. 2024b. Magpie: Alignment Data Synthesis from Scratch by Prompting Aligned LLMs with Nothing. *ArXiv*, abs/2406.08464.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; Lin, H.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Lin, J.; Dang, K.; Lu, K.; Bao, K.; Yang, K.; Yu, L.; Li, M.; Xue, M.; Zhang, P.; Zhu, Q.; Men, R.; Lin, R.; Li, T.; Xia, T.; Ren, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Qiu, Z. 2024. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*.

Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Zhang, D.; Dai, Q.; and Peng, H. 2025. The Best Instruction-Tuning Data are Those That Fit.

Zhang, J.; Qin, Y.; Pi, R.; Zhang, W.; Pan, R.; and Zhang, T. 2024. TAGCOS: Task-agnostic Gradient Clustered Coreset Selection for Instruction Tuning Data. *ArXiv*, abs/2407.15235.

Zhou, C.; Liu, P.; Xu, P.; Iyer, S.; Sun, J.; Mao, Y.; Ma, X.; Efrat, A.; Yu, P.; Yu, L.; Zhang, S.; Ghosh, G.; Lewis, M.; Zettlemoyer, L.; and Levy, O. 2023a. LIMA: Less Is More for Alignment. *ArXiv*, abs/2305.11206.

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

Zhou, J.; Lu, T.; Mishra, S.; Brahma, S.; Basu, S.; Luan, Y.; Zhou, D.; and Hou, L. 2023c. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.