CrowdSelect: Synthetic Instruction Data Selection with Multi-LLM Wisdom

Anonymous ACL submission

Abstract

Distilling advanced Large Language Models' instruction-following capabilities into smaller models using a selected subset has become a mainstream approach in model training. While existing synthetic instruction data selection strategies can identify valuable subsets for distillation, they predominantly rely on single-dimensional signals (*i.e.*, reward scores, model perplexity). We argue that such narrow signals may overlook essential nuances of user instructions, especially when each instruction can be answered from multiple perspectives. Therefore, we investigate more diverse signals to capture comprehensive instruction-response pair characteristics and propose three foundation metrics that leverage Multi-LLM wisdom: *(1)* diverse responses across multiple LLMs and *(2)* reward model assessment. Based on these metrics, we propose CrowdSelect, which combines all three metrics with diversity preservation through clustering. Our comprehensive experiments demonstrate that our foundation metrics consistently improve performance across 4 base models on MT-bench and Arena-Hard. Our CrowdSelect, as an integrated metric, achieves *state-of-the-art* performance in both Full and LoRA fine-tuning, showing improvements of 4.81% on Arena-Hard and 11.1% on MT-bench with Llama-3.2-3b-instruct. We hope our findings will bring valuable insights for future research in this direction.

1 Introduction

In recent years, Large Language Models (LLMs) (Achiam et al., 2023; Jaech et al., 2024; Team et al., 2024; Guo et al., 2025) have demonstrated remarkable capability in following user instructions to generate coherent and contextually helpful responses (Jiang et al., 2023; Zheng et al., 2023b; Wen et al., 2024). Yet, the computational overhead for instruction tuning and massive parameter sizes of these models create a considerable barrier to practical deployment (Peng et al., 2023). To address this, many approaches distill the instruction-following prowess of advanced LLMs into smaller, more efficient models through a small-amount instruction tuning with synthetic responses (Xia et al., 2024; Zhou et al., 2024a).

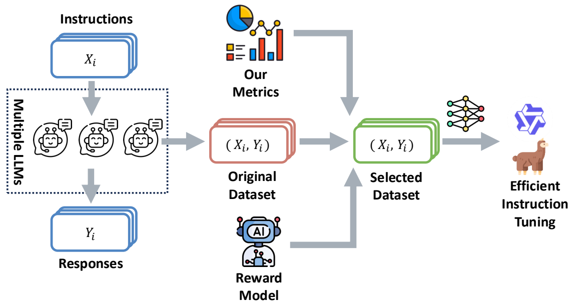

Figure 1: A demonstration of instruction tuning with selected synthetic instruction-response pairs.

A critical bottleneck, however, lies in selecting the right data for this distillation process. Most existing data selection methods rely on predefined rules (Chen et al., 2023a), automated single-dimensional signals — such as reward scores (Wu et al., 2024b; Lambert et al., 2024) or difficulty metrics (Li et al., 2023b, 2024b) — to identify valuable examples for fine-tuning. While effective to a point, such narrow signals may overlook essential nuances of user instructions, especially when each instruction can be answered from multiple perspectives (Händler, 2023; Feng et al., 2025). This raises a fundamental question: *"Can we leverage multi-dimensional signals to better reflect the various facets of each sample for more effective instruction tuning data selection?"*

Inspired by previous work that leverage Multi-LLMs collaboration, we take a explorative step toward more robust and comprehensive data selection by introducing CrowdSelect, a framework that harnesses pre-collected Multiple LLMs' responses and their reward scores, treating as different reflection of the instruction to leverage Multi-LLM Wisdom. Instead of treating each instruction–response pair in isolation—typically derived from just one model's output—our method aggregates multiple responses for each instruction from a diverse set of LLMs. Crucially, we also factor in each response's reward as provided by state-of-the-art reward models. This multi-view setup captures more *"facets"* of each instruction, illuminating subtle differences in how various models handle the same query. Based on these observations, we propose three base explorative metrics:

- **Difficulty -** Identifies instructions on which the majority of models struggle, surfacing challenging prompts critical to learning.
- **Separability –** Highlights instructions whose response quality exhibits high variance across models, making them especially useful for differentiating stronger from weaker capabilities.
- **Stability –** Measures how consistently model performance follows expected size-based ranking across families, ensuring the selected data helps reinforce well-grounded alignment signals.

Our exploratory experiments in FFT and low-rank adaptation (LoRA) (Hu et al., 2021) experiments on LLama-3.2-3b-base/instruct (Dubey et al., 2024) and Qwen-2.5-3b-base/instruct (Yang et al., 2024b) demonstrate the robustness and efficacy of our proposed metrics through significant performance gaps between *top-scored* and *bottom-scored* data subset fine-tuning, with potential further improvements through metric combination.

Subsequently, we propose CROWDSELECT that combines these metrics with a clustering strategy to preserve diversity and exlore the upperbound of leveraging Multi-LLM wisdom to identify a compact yet high-impact subset of instruction-response data. Experimental results show that models fine-tuned on our selected subset significantly outperform baselines and previous state-of-the-art data selection methods, achieving improvements of 4.81% on Arena-Hard and 11.1% on MT-bench with Llama-3.2-3b-instruct. Furthermore, CROWDSELECT achieves *state-of-the-art* performance across four models on two benchmarks, demonstrating both the generalizability and robustness of our selected data and methodology, paving a new dimension for efficient instruction tuning.

Our contributions are summarized as follows:

- **Investigation of Multi-LLM Wisdom in Instruction Data Selection.** We propose a novel approach that utilizes multiple synthesized responses from different LLMs for each instruction, enhancing the diversity and quality of data.
- **Novel Metrics and Methods.** We design three new explorative base metrics—*Difficulty*, *Separability*, and *Stability*—that leverage multi-LLM responses and reward scores as more comprehensive signals, and combine them into CROWDSELECT to explore upperbound in selecting high-quality data for instruction tuning.
- ***State-of-the-art* Performance.** We demonstrate that combining our metrics and clustering techniques for data selection leads to a new SOTA in efficient instruction tuning in both Llama-3.2-3b and Qwen-2.5-3b.

2 Related Work

Instruction Tuning Data Selection. Instruction Tuning stands out to be a method to solve the gap between pretrained knowledge and real-world user scenarios (Ouyang et al., 2022; Bai et al., 2022). Recent efforts like Vicuna (Peng et al., 2023) and LIMA (Zhou et al., 2024a) demonstrate high performance with a carefully selected small dataset, highlighting the growing importance of efficient instruction tuning. Three key metrics determine instruction data quality: *Difficulty*, *Quality*, and *Diversity*. *Difficulty*, focusing mainly on the question side, is considered more valuable for model learning (Li et al., 2023b, 2024b; Liu et al., 2024b; Lee et al., 2024; Wang et al., 2024b). *Quality*, mainly addressing the response side, measures the helpfulness and safety of model responses, typically assessed using LLM evaluators (Chen et al., 2023a, 2024b; Liu et al., 2024c; Ye et al., 2024), reward models (Son et al., 2024; Lambert et al., 2024), and gradient similarity search (Xia et al., 2024). *Diversity* also plays a crucial role in covering various instruction formats and world knowledge, primarily improving model robustness (Bukharin and Zhao, 2023; Wang et al., 2024d).

Data Synthesis for Instruction Tuning. While the development of LLMs initially relied on human-curated instruction datasets for instruction tuning (Zheng et al., 2023a; Zhao et al., 2024; Lightman et al., 2023), this approach proved time-consuming and labor-intensive, particularly as the complexity and scope of target tasks increased (Demrozi et al., 2023; Wang et al., 2021). Consequently, researchers began exploring the use of frontier LLMs to generate synthetic instruction datasets, aiming to both address these scalability challenges (Ding et al., 2023; Chen et al., 2023b, 2024d) and leverage models' advanced capabilities in developing next-generation foundation models (Burns et al., 2023; Charikar et al., 2024). Recent advancements streamline this process by utilizing instructions directly from pretrained LLMs with simple prompt templates (Xu et al., 2024a; Chen et al., 2024c; Zhang et al., 2024), significantly reducing the required custom design from human effort.

Deriving Crowded Wisdom from Multi-LLM. Single LLM's response to a question face limitations in its representation of data (particularly cutting-edge knowledge) (Lazaridou et al., 2021; Dhingra et al., 2022; Kasai et al., 2023), skills (as no single LLM is universally optimal *empirically*) (Sun et al., 2022; Liang et al., 2022; Chen et al., 2024a), and diverse perspectives (Feng et al., 2025). Previous work has demonstrated that *online* multi-LLM wisdom (also known as compositional agent frameworks (Gupta and Kembhavi, 2023)) tends to outperform single models across various domains, providing more comprehensive and reflective solution on complex downstream tasks (Wang et al., 2024c; Wu et al., 2023; Li et al., 2023a; Ouyang et al., 2025; Gui et al., 2025). *Offline* crowded wisdom, where data are pre-collected rather than real-time inference, also show potential in model alignment (Gallego, 2024; Rafailov et al., 2023; Meng et al., 2025) and benchmark construction (Ni et al., 2024b,a). In this paper, we pioneer the use of *offline* multi-LLM wisdom for instruction data selection by utilizing these LLMs' responses and their reward score as *reflections* to measure instruction-response pairs' *Difficulty* and *Quality*.

3 Methodology

In this section, we first define our synthetic data selection task and propose three foundational metrics that leverage responses and assessment scores from multiple advanced LLMs. We then introduce CrowdSelect, which combines these metrics with diversity-preserving clustering to explore the upper bounds of Multi-LLM Wisdom.

3.1 Preliminaries

We formulate the instruction quality as the consensus among N LLMs. Given an instruction-tuning dataset, we extract all instructions from the dataset to form instruction dataset Q. For each instruction $q_i \in Q$, a response set R_i is obtained by querying multiple LLMs. An assessment model then evaluates the response set R_i to form the score set C_i^M based on metrics M. The index of M is omitted for brevity in the following context unless specified. The top-k instruction subset of metric M is defined as

$$S_k^M = \underset{S \subset \mathcal{S}, |S|=k}{\arg\max}\, M(C_i^M) \tag{1}$$

where S_k^M consists of the k instructions that maximize the metric M.

The corresponding response r_i^M for each instruction q_i^M from the instruction subset S_k^M is subsequently obtained by

$$r_i^M = \mathrm{Top}(R_i, C_i^M) \tag{2}$$

where $\mathrm{Top}(R_i^S, C_i^M)$ denotes the best responses in r_i^S ranked by C_i^M. The produced instruction-answer subset $\hat{Q} = \{(r_i^M, q_i^M)\}$ is then utilized for finetuning as an alternative of the original dataset.

3.2 Base Metrics

In this section, we introduce three new base metrics to leverage multiple LLMs' responses and their reward scores as various *"facets"* to reflect the value of each sample.

Difficulty. The difficulty score C^{diff} is defined as the negative value of the average score, which is the mean score of all the model responses for a given instruction.

$$C^{diff} = -\frac{\sum C_i^M}{N} \tag{3}$$

Higher *difficulty* indicates more challenging instructions. This metric is particularly well-suited for fine-tuning on reasoning tasks, e.g. mathematics and planning, where the goal is often to improve performance on complex problems. By focusing on instructions with higher *difficulty*, we prioritize examples that are likely to be answered incorrectly by the majority of models. This ensures that the fine-tuning dataset includes a substantial proportion of challenging instructions, maximizing the model's exposure to difficult material and potentially leading to greater improvements in performance.

Separability. The separability score C^{sep} is defined as the score variance, which is the variance of all the response scores for an instruction.

$$C^{sep} = \mathrm{var}(C_i^M) \tag{4}$$

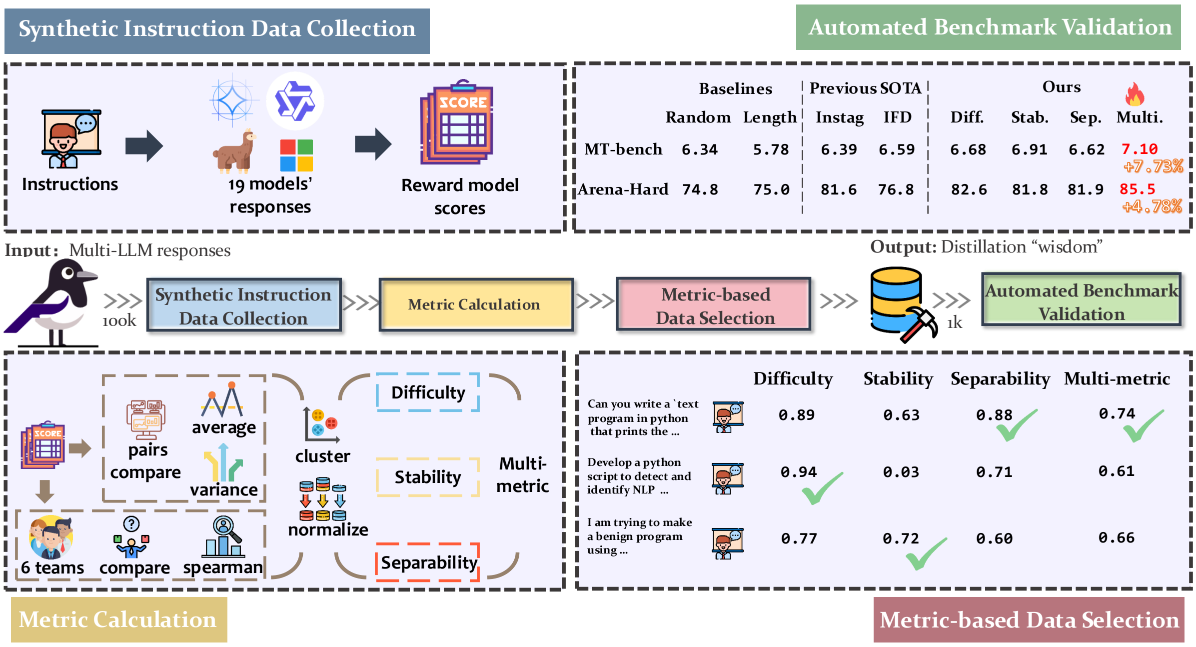

Figure 2: The overall pipeline of our CROWDSELECT, which innovatively leverages metrics calculated from multiple facets of instructions using pre-collected synthesized responses from various LLMs and their corresponding reward model scores. We enhance data selection through clustering for diversity and metric combination to explore the method's potential. Finally, we evaluate the effectiveness of our selected instruction subset through FFT or LoRA fine-tuning (Hu et al., 2021) for efficient instruction tuning.

Higher *Separability* indicates that a considerable proportion of models cannot perform well on the instruction, thus this instruction is more effective in differentiating between models. This characteristic makes the *Separability* particularly well-suited for curating datasets of knowledge remembering or preference alignment. In such datasets, some models may exhibit strong performance while others struggle. By selecting instructions with high separability, we prioritize examples that effectively distinguish between these varying levels of competence. These *"discriminatory"* examples are valuable because they provide the fine-tuned model with opportunities to learn from the specific challenges that differentiate successful models from less successful ones. Focusing on these examples enforces the finetuned model to handle the nuances and complexities that separate high-performing models.

Stability. *Stability* is defined as the average spearman factor, which is the mean of five spearman factors, corresponding to five model families. The spearman factor is calculated based on r^a and r^b:

$$\frac{\frac{1}{n}\sum_{i=1}^{n}(r_i^a-\overline{r^a})\cdot\left(r_i^b-\overline{r^b}\right)}{\sqrt{\left(\frac{1}{n}\sum_{i=1}^{n}\left(r_i^a-\overline{r^a}\right)^2\right)\cdot\left(\frac{1}{n}\sum_{i=1}^{n}\left(r_i^b 2-\overline{r^b}\right)^2\right)}} \tag{5}$$

- r^a refers to the original ranking within a model family, where models with larger parameters are theoretically ranked higher, naturally aligning with the performance rank.
- r^b is determined by the ranking of models based on their response scores (*e.g.*, if LLaMA-3B has a response score of 90 and LLaMA-8B has a response score of 75, then 3B ranks higher than 8B within the LLaMA family).

Stability effectively captures how well performance rankings align with expected model size rankings using Spearman's rank correlation (Schober et al., 2018), making it robust to variations in score scales and non-linear relationships. Averaging across model families further strengthens the robustness of the score, alleviating performance gaps among model families.

3.3 CROWDSELECT: Explore the Upperbound with Multi-LLM Wisdom

Diversity Preservation with Clustering. To facilitate clustering, all instructions were embedded into a fixed-dimensional latent space using a pre-trained embedding model. Within each cluster, instructions were then ranked with the given metric, and the highest-ranked instructions were selected. To avoid over-representing dominant clusters and neglecting potentially valuable information contained within smaller or less frequent clusters, we draw equally from each cluster to form a more robust and generalizable subset.

Multi-metric Integration Building upon the cluster-based selection strategy, we introduce a multi-metric approach to leverage the diverse information captured by the difficulty, separability, and stability scores. Each instruction-response pair is thus characterized by a vector of associated scores, reflecting its various attributes. However, these metrics exhibit different distributions, ranges, and magnitudes. Therefore, we employ a three-stage normalization process to ensure equitable contribution from each metric.

Specifically, each metric score is standardized to standard normal distribution. The standardized scores are then normalized to $[0, 1]$ using a min-max scaling approach. Finally, to further refine the distribution and mitigate the impact of potential outliers, we apply a quantile transformation that maps the normalized scores to a uniform distribution between $[0, 1]$.

$$Z_i^M = \frac{(C_i^M - \mu^M)}{\sigma^M} \tag{6}$$

$$N_i^M = \frac{(Z_i^M - min(Z^M))}{(max(Z^M) - min(Z^M))} \tag{7}$$

$$\rho_i^M = \text{quant}(N_i^M | N^M) \tag{8}$$

Following this normalization procedure, we aggregate the transformed scores into a single multi-metric score $\hat{C}$ for each instruction-response pair. This aggregation is performed using a weighted sum of the proposed metrics:

$$\hat{C}_i = \sum_j w_i * \rho_i^{M_j} \tag{9}$$

where $\rho_i^{M_j}$ represent the quantile-transformed scores for metric j, and w_i are the corresponding weights assigned to each metric. This weighted multi-metric approach, combined with the preceding normalization steps, ensures a balanced and robust data selection process that leverages the complementary information provided by the different metrics.

4 Experiment

In this section, we first validate our base metrics through comparative experiments between top-scored and bottom-scored data subsets. We then evaluate CROWDSELECT against existing baselines and *state-of-the-art* methods. Finally, we conduct an ablation study to analyze the contribution of each sub-module within CROWDSELECT.

4.1 Experiment Setups

Datasets. We conduct our experiments on Magpie-100K-Generator-Zoo[1] given that it directly matches our problem setting that contains answers from 19 models—Qwen2 (Yang et al., 2024a), Qwen2.5 (Yang et al., 2024b), Llama 3 (Dubey et al., 2024), Llama 3.1 (Dubey et al., 2024), Gemma 2 (Team et al., 2024), Phi-3 (Abdin et al., 2024) families and GPT-4(Achiam et al., 2023)—and their reward scores from three state-of-the-art reward models from RewardBench (Lambert et al., 2024): ArmoRM-Llama3-8B-v0.1 (Wang et al., 2024a), Skywork-Reward-Llama-3.1-8B (Liu and Zeng, 2024), and Skywork-Reward-Gemma-2-27B (Liu and Zeng, 2024).

Evaluation. To evaluate the instruction-following capabilities, we use two widely-used instruction-following benchmarks: MT-Bench (Zheng et al., 2023b) and Arena-Hard (Li et al., 2024c). Both benchmarks mainly leverage LLM-as-a-Judge (Zheng et al., 2023b) for evaluation, while MT-Bench leverage 1-10 rating scoring and Arena-Hard leverage direct pairwise comparison and finally provide a leaderboard with one model as anchor-points. In our experiment, we set the base model (*i.e.*, LLaMA-3.2-3B-base) as the anchor point for models for arena battles. We unified the LLM-as-a-Judge model in both benchmarks as DeepSeek-V3 (Liu et al., 2024a) through official API[2] and Together API[3] given its high performance on natural language generation tasks. Thanks to the unified judge model, we additionally report the **Average Performance (AP)** as a ranking computed by the ranking in MT-Bench and Arena-Hard. **Each experiment was conducted 3 times. The average results are reported to ensure the reliability and reproducibility.**

Base Models. Following Xu et al. (2024b), we consider four small models from different developers as student models, including base

[1] https://huggingface.co/datasets/Magpie-Align/Magpie-100K-Generator-Zoo

[2] https://platform.deepseek.com/

[3] https://api.together.ai/

Table 1: Validation of our three foundation metrics on Full fine-tuning Llama-3b-instruct with *top-scored (↑)* and *bottom-scored (↓)* instruction selection and different response selection strategy. Best and second results for each metric is highlighted in **bold** and underline.

Strategy	DirectScore	Difficulty ↓	Difficulty ↑	Separability ↓	Separability ↑	Stability ↓	Stability ↑	Multi
				MT-Bench				
Best-answer	4.406	4.506	4.738	4.731	**5.056**	4.675	**5.088**	**5.125**
Random	**4.470**	4.469	4.688	4.695	4.785	4.500	4.581	4.613
Top5-random	4.435	4.681	**4.870**	4.788	5.008	4.619	4.956	5.048
				Arena-Hard				
Best-answer	**75.3$_{(-2.0, 1.6)}$**	78.6$_{(-1.9, 2.1)}$	76.8$_{(-1.6, 1.7)}$	81.8$_{(-1.8, 1.2)}$	**83.3$_{(-1.8, 1.7)}$**	80.0$_{(-1.5, 1.6)}$	**82.3$_{(-1.6, 2.2)}$**	**85.5$_{(-0.8, 1,1)}$**
Random	74.5$_{(-1.1, 1.2)}$	78.5$_{(-1.6, 1.3)}$	**80.4$_{(-1.0, 1.5)}$**	79.0$_{(-1.3, 1.4)}$	80.6$_{(-1.6, 1.6)}$	76.2$_{(-0.8, 1.6)}$	77.0$_{(-1.0, 1.8)}$	82.3$_{(-1.2, 1.3)}$
Top5-random	73.7$_{(-1.2, 1.8)}$	75.9$_{(-1.6, 1.5)}$	76.8$_{(-1.2, 1.4)}$	82.0$_{(-1.3, 1.2)}$	80.0$_{(-0.7, 1.3)}$	75.0$_{(-4.4, 5.8)}$	76.9$_{(-1.4, 1.6)}$	83.1$_{(-1.4, 1.7)}$

and instruct models—Qwen-2.5-3B, Qwen-2.5-3B-Instruct (Yang et al., 2024b) and LLaMA-3.2-3B, LLaMA-3.2-3B-Instruct (Dubey et al., 2024). We use 10 clusters for diversity preservation, and the multimetric setting uses $w = (1, 1, 2)$ for metric intergration in the following experiments.

Baselines. We include 7 baselines in our experiments. *Random*, denotes a randomly selected instruction-answer set from the original dataset. We also compared two previous *state-of-the-art* data selection method: Instag (Lu et al., 2023) , and IFD (Li et al., 2023b). For rule-based method, We include *Length* and *Reward Score* (Liu et al., 2023). More details are shown in Appendix B.3.

Instruction-Tuning Setups. We conduct our fine-tuning and evaluation on single A800 and A6000 servers. For fine-tuning, we use LLaMA-Factory (Zheng et al., 2024). For evaluation, we leverage the official codebase of MT-Bench[4] and Arena-Hard[5] for automatic assessments. See Appendix B for more details of experiment setups.

4.2 Experiment Results.

Three foundation metrics demonstrate effectiveness in selecting valuable samples. As shown in Table 1, our three foundation metrics consistently identify valuable instruction samples across all response selection strategies. Models fine-tuned on *Top-scored* samples consistently outperform *Bottom-scored* samples, with *Stability* exceed the most margin. We also explore the response selection strategies to build a foundation for following experiments. *Best-answer* setting outperforms both *Random* and *Top5-random* approaches, indicating that responses with higher reward scores provide better quality data for distillation. This consistent performance across individual metrics establishes strong foundation for further improvements through integration. Therefore, we use *top-scored* as the instruction selection and *Best-answer* as the corresponding response for all experiments.

CrowdSelect achieves new *state-of-the-art* performance on both benchmarks. As shown in Table 2, our approach significantly outperforms previous baselines across four models, demonstrating robust generalization. On Arena-Hard and MT-bench, CrowdSelect with Llama-3.2-3b-instruct achieves scores of 85.5 and 7.103 respectively, surpassing the previous best results by 4.81% and 11.1%. For Qwen-2.5-3b-instruct, CrowdSelect outperforms the strongest baseline by 3.90%, validating our approach of post-training with high-quality instructions and model distillation. Even for base models, our foundation metrics and CrowdSelect prove effective, notably improving Llama-3.2-3b's performance on MT-bench by 12.3%.

CrowdSelect metrics perform robust on various finetuning methods. Beyond demonstrating superior performance on standard benchmarks, the proposed metrics were further evaluated for robustness across a range of fine-tuning methodologies. Table 1 revealed consistent and stable performance of the proposed metrics. This robustness across varying training paradigms highlights the generalizability of the metrics and suggests their applicability in a wider range of practical scenarios.

4.3 Ablation Studies

In this section, we conduct ablation studies for each module in CrowdSelect to provide a compre-

[4] https://github.com/lm-sys/FastChat/tree/main/fastchat/llm_judge

[5] https://github.com/lmarena/Arena-Hard-auto

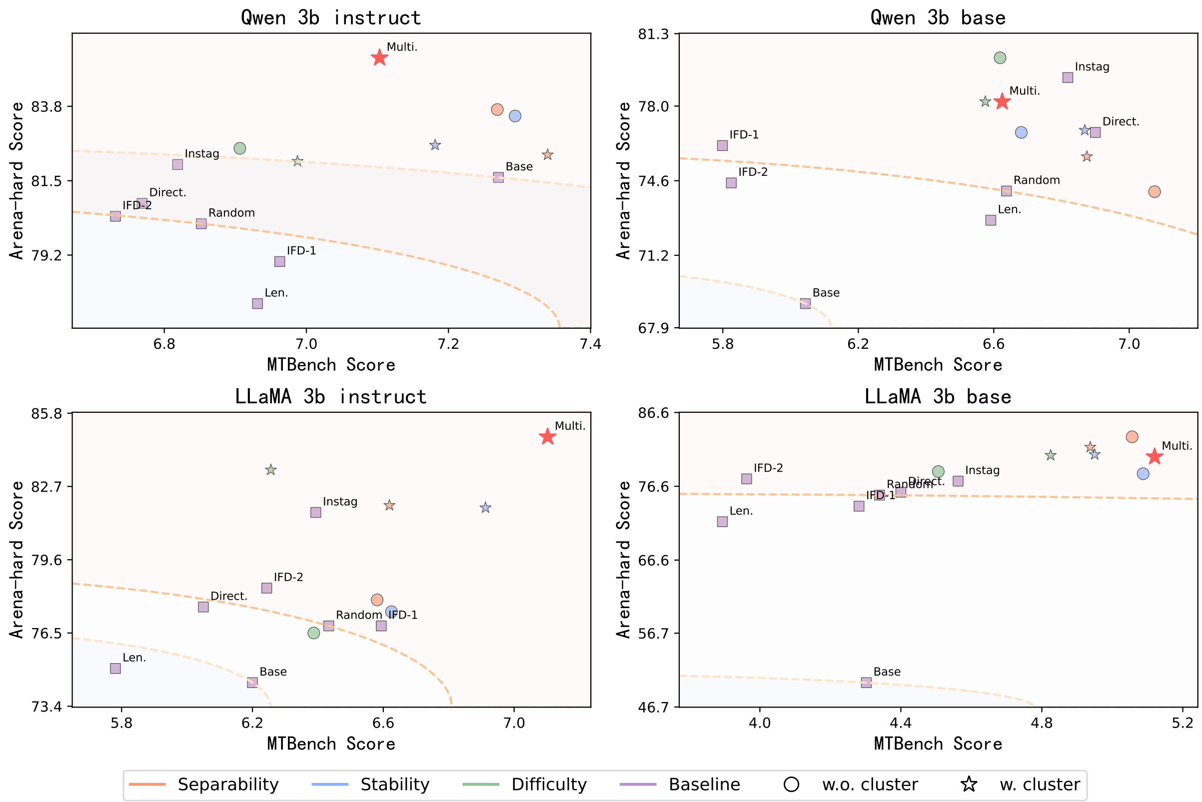

Figure 3: Overall results demonstrate that our foundation metrics and CROWDSELECT consistently outperform baseline methods by a significant margin across FFT settings of four models, with particularly strong performance improvements on Llama-3b-instruct.

hensive analysis of our approach.

Dataset Size. Cao et al. (2023) suggests that selecting concise subsets from all datasets yield competitive results. Following this finding, we collect 1k instruction-response pairs overall in our main experiments. Further experiments on various dataset sizes also support this finding. From the results in 4, small elite datasets behaves on par with a large dataset. This highlights the importance of data quality over sheer quantity in instruction tuning.

Metric Coefficient Combination Our experiments explored various coefficient combinations to determine the optimal balance for creating high-quality, robust datasets. Table 3 details the process of optimizing the weights assigned to different metrics when evaluating dataset quality. As shown in the table, the coefficient combination w = (1, 1, 2) consistently yielded superior results compared to other tested combinations.

Number of Clusters Clustering's impact on dataset quality was investigated by varying the number of clusters during dataset construction (see Table 4). While no strong positive correlation was observed between cluster count and quality, all clustered datasets outperformed those constructed without clustering. The results highlight the importance and robustness of the clustering process.

Response Generation Strategy The response generation strategy largely affects the generation quality of the finetuned LLM. Table 1 shows that the best-answer strategy substantially outperforms other strategies, highlighting the importance of dataset response quality. We argue the reason for strategy independence of the difficulty metric is that the core challenge in these instructions is inherently tied to the complexity of the task, not in the method of response formulation. For instance, a highly challenging instruction may require the model to synthesize information from multiple domains, reason through abstract concepts, or produce detailed, contextually rich outputs. These demands remain consistent, regardless of response generation strategies.

Further experiments on finetuning with LoRA and reward model selection are also presented in Appendix C.

Table 2: Performance comparison of full finetuned Llama3.2-3b-base/instruct and Qwen2.5-3b-base/instruct models with different data selection strategies. The best and second results are in **bold** and underline.

Benchmark	Base	Baselines			Our Metrics			
		Random	Tags	IFD	Difficulty	Separability	Stability	Multi
Llama3.2-3b-base								
MT-Bench	4.302	4.406	4.562	3.962	4.738	5.056	5.088	**5.125**
Arena-Hard	50.0(-0.0, 0.0)	75.3(-2.0, 1.6)	77.3(-1.1, 1.2)	77.6(-1.6, 1.6)	76.8(-1.6, 1.7)	**83.3(-1.8, 1.7)**	78.3(-1.6, 2.2)	80.6(-2.4, 1.6)
Llama3.2-3b-instruct								
MT-Bench	6.200	6.356	6.393	6.243	6.648	6.581	6.625	**7.103**
Arena-Hard	74.4(-1.0, 1.5)	74.8(-1.5, 1.6)	81.6(-0.2, 0.2)	78.4(-1.7, 1.5)	80.5(-0.9, 1.3)	77.9(-1.5, 1.7)	77.4(-1.5, 1.1)	**85.5(-0.8, 1.1)**
Qwen2.5-3b-base								
MT-Bench	6.043	6.500	6.818	5.825	6.613	**7.075**	6.681	6.625
Arena-Hard	69.0(-2.2, 1.6)	72.9(-2.2, 1.9)	79.3(-2.2, 1.9)	74.5(-1.5, 1.5)	73.8(-2.5, 1.8)	74.1(-1.6, 2.4)	76.8(-1.8, 1.8)	**79.9(-1.6,1.8)**
Qwen2.5-3b-instruct								
MT-Bench	7.138	6.793	6.818	6.731	7.182	7.269	**7.294**	7.131
Arena-Hard	81.6(-1.8, 1.4)	78.2(-1.7, 2.0)	82.0(-2.4, 1.6)	80.4(-1.3, 1.0)	81.8(-1.6, 1.3)	83.7(-1.4, 1.2)	83.5(-1.4, 1.4)	**85.2(-1.2, 1.1)**

Table 3: Hyperparameter comparison of CROWDSELECT using Llama-3b-instruct models with varying cluster numbers. The sequence represents (*Difficulty*, *Separability*, *Stability*).

Hyperparameter	MT-Bench	Arena-Hard
1_1_1	6.913	81.8(-0.5, 0.8)
1_-1_1	6.625	84.2(-0.7, 1.0)
1_1_2	**7.103**	**85.5**(-0.8, 1.1)
1_1_-1	6.650	82.7(-1.5, 1.4)
1_1_1.5	6.850	84.7(-1.6, 1.3)
1_-1_1.5	6.781	83.0(-1.4, 1.4)
-1_-1_1	6.781	81.9(-1.5, 1.3)
-1_-1_2	6.838	84.8(-1.3, 1.2)
-1_-1_1.5	6.638	81.8(-1.3, 1.3)

Table 4: Performance comparison of FFT-version of Llama-3b-instruct on different coefficient combinations for multiple metrics with clustering.

Benchmark	Random	Difficulty	Separability	Stability
10 clusters				
MT-Bench	6.443	6.675	6.619	**6.913**
Arena-Hard	80.9	82.6	81.9	81.8
Arena-Hard-95%CI	(-1.3, 1.4)	(-1.2, 1.8)	(-1.7, 1.7)	(-1.5, 1.7)
20 clusters				
MT-Bench	6.607	6.615	6.591	**6.686**
Arena-Hard	82.8	83.1	**85.2**	82.8
Arena-Hard-95%CI	(-1.2, 1.4)	(-1.1, 1.7)	(-1.3, 1.1)	(-1.4, 1.1)
30 clusters				
MT-Bench	6.721	6.737	6.725	6.562
Arena-Hard	83.2	**84.9**	83.3	83.8
Arena-Hard-95%CI	(-1.3, 1.1)	(-1.0, 1.1)	(-1.4, 1.4)	(-1.4, 1.2)

Figure 4: Results show that small elite datasets behaves on par with a large dataset. Our implementation (line in red) achieves reasonably good results on all scenarios.

5 Conclusion

This paper presents novel metrics for synthetic instruction data selection based on Multi-LLM Wisdom, capturing the *difficulty* of instructions from multiple perspectives through various LLMs' responses and their corresponding reward scores. We validate our hypothesis through the strong performance of individual metrics on both MT-Bench and Arena-Hard using FFT and LoRA fine-tuning on Llama-3.2-3b and Qwen-2.5-3b. By combining diversity enhancement through clustering with our proposed metrics, CROWDSELECT consistently outperforms *state-of-the-art* data selection methods, establishing both new perspectives and a robust baseline for instruction tuning data selection.

Limitations

While CROWDSELECT demonstrates significant improvement in synthetic data selection tasks, we acknowledge several limitations. Our approach computes data selection metrics by leveraging responses from multiple models across different model families and their corresponding reward model scores. However, this methodology may be susceptible to reward model biases, including potential reward hacking issues. Although a more organic integration of multiple reward scores could potentially enhance robustness, the computation of these scores requires additional computational resources. Furthermore, our experiments were conducted on both A800 and A6000 GPUs, and the variation in hardware environments may introduce some instability and affect experimental results, potentially impacting the reproducibility of our findings.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. 2024. A survey on data selection for language models. *ArXiv*, abs/2402.16827.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Alexander Bukharin and Tuo Zhao. 2023. Data diversity matters for robust instruction tuning. *arXiv preprint arXiv:2311.14736*.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas R. Joglekar, Jan Leike, Ilya Sutskever, Jeff Wu, and OpenAI. 2023. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *ArXiv*, abs/2312.09390.

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. 2023. Instruction mining: Instruction data selection for tuning large language models. *arXiv preprint arXiv:2307.06290*.

Moses Charikar, Chirag Pabbaraju, and Kirankumar Shiragur. 2024. Quantifying the gain in weak-to-strong generalization. *ArXiv*, abs/2405.15116.

Dongping Chen, Ruoxi Chen, Shu Pu, Zhaoyi Liu, Yanru Wu, Caixi Chen, Benlin Liu, Yue Huang, Yao Wan, Pan Zhou, et al. 2024a. Interleaved scene graph for interleaved text-and-image generation assessment. *arXiv preprint arXiv:2411.17188*.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yinuo Liu, Yaochen Wang, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. 2024b. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. *arXiv preprint arXiv:2402.04788*.

Jiuhai Chen, Rifaa Qadri, Yuxin Wen, Neel Jain, John Kirchenbauer, Tianyi Zhou, and Tom Goldstein. 2024c. Genqa: Generating millions of instructions from a handful of prompts. *ArXiv*, abs/2406.10323.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023a. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Lin Chen, Jinsong Li, Xiao wen Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2023b. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*.

Lin Chen, Xilin Wei, Jinsong Li, Xiao wen Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Bin Lin, Zhenyu Tang, Li Yuan, Yu Qiao, Dahua Lin, Feng Zhao, and Jiaqi Wang. 2024d. Sharegpt4video: Improving video understanding and generation with better captions. *ArXiv*, abs/2406.04325.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *ArXiv*, abs/2310.01377.

Florenc Demrozi, Cristian Turetta, Fadi Al Machot, Graziano Pravadelli, and Philipp H. Kindt. 2023. A comprehensive review of automated data annotation techniques in human activity recognition. *ArXiv*, abs/2307.05988.

Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *ArXiv*, abs/2305.14233.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yann Dubois, Bal'azs Galambosi, Percy Liang, and Tatsunori Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *ArXiv*, abs/2404.04475.

Shangbin Feng, Wenxuan Ding, Alisa Liu, Zifeng Wang, Weijia Shi, Yike Wang, Zejiang Shen, Xiaochuang Han, Hunter Lang, Chen-Yu Lee, Tomas Pfister, Yejin Choi, and Yulia Tsvetkov. 2025. When one llm drools, multi-llm collaboration rules.

Víctor Gallego. 2024. Refined direct preference optimization with synthetic data for behavioral alignment of llms. *arXiv preprint arXiv:2402.08005*.

Chujie Gao, Qihui Zhang, Dongping Chen, Yue Huang, Siyuan Wu, Zhengyan Fu, Yao Wan, Xiangliang Zhang, and Lichao Sun. 2024. The best of both worlds: Toward an honest and helpful large language model. *arXiv preprint arXiv:2406.00380*.

Yi Gui, Yao Wan, Zhen Li, Zhongyi Zhang, Dongping Chen, Hongyu Zhang, Yi Su, Bohua Chen, Xing Zhou, Wenbin Jiang, et al. 2025. Uicopilot: Automating ui synthesis via hierarchical code generation from webpage designs. In *THE WEB CONFERENCE 2025*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14953–14962.

Thorsten Händler. 2023. Balancing autonomy and alignment: A multi-dimensional taxonomy for autonomous llm-powered multi-agent architectures. *ArXiv*, abs/2310.03659.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. ORPO: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189, Miami, Florida, USA. Association for Computational Linguistics.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. 2023. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Yue Huang, Siyuan Wu, Chujie Gao, Dongping Chen, Qihui Zhang, Yao Wan, Tianyi Zhou, Xiangliang Zhang, Jianfeng Gao, Chaowei Xiao, et al. 2024. Datagen: Unified synthetic dataset generation via large language models.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2023. Followbench: A multi-level fine-grained constraints following benchmark for large language models. *arXiv preprint arXiv:2310.20410*.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, Kentaro Inui, et al. 2023. Realtime qa: What's the answer right now? *Advances in neural information processing systems*, 36:49025–49043.

Jingun Kwon, Hidetaka Kamigaito, Manabu Okumura, et al. 2024. Instructcmp: Length control in sentence compression through instruction-based large language models. *arXiv preprint arXiv:2406.11097*.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. 2024. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*.

Angeliki Lazaridou, Adhi Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, et al. 2021. Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems*, 34:29348–29363.

Changho Lee, Janghoon Han, Seonghyeon Ye, Stanley Jungkyu Choi, Honglak Lee, and Kyunghoon Bae. 2024. Instruction matters, a simple yet effective task selection approach in instruction tuning for specific tasks. *arXiv preprint arXiv:2404.16418*.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008.

Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, Yuxian Gu, Xin Cheng, Xun Wang, Si-Qing Chen, Li Dong, Wei Lu, Zhifang Sui, Benyou Wang, Wai Lam, and Furu Wei. 2024a. Synthetic data (almost) from scratch: Generalized instruction tuning for language models. *ArXiv*, abs/2402.13064.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024b. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In *Annual Meeting of the Association for Computational Linguistics*.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2023b. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. *arXiv preprint arXiv:2308.12032*.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024c. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*.

Zhuang Li, Yuncheng Hua, Thuy-Trang Vu, Haolan Zhan, Lizhen Qu, and Gholamreza Haffari. 2024d. Scar: Efficient instruction-tuning for large language models via style consistency-aware response ranking. *ArXiv*, abs/2406.10882.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *ArXiv*, abs/2305.20050.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Chris Yuhao Liu and Liang Zeng. 2024. Skywork reward model series. `https://huggingface.co/Skywork`. Hugging Face model repository.

Liangxin Liu, Xuebo Liu, Derek F Wong, Dongfang Li, Ziyi Wang, Baotian Hu, and Min Zhang. 2024b. Selectit: Selective instruction tuning for large language models via uncertainty-aware self-reflection. *arXiv preprint arXiv:2402.16705*.

Liangxin Liu, Xuebo Liu, Derek F. Wong, Dongfang Li, Ziyi Wang, Baotian Hu, and Min Zhang. 2024c. Selectit: Selective instruction tuning for large language models via uncertainty-aware self-reflection. *ArXiv*, abs/2402.16705.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2023. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *ArXiv*, abs/2312.15685.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. # instag: Instruction tagging for analyzing supervised fine-tuning of large language models. In *The Twelfth International Conference on Learning Representations*.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. Wizardcoder: Empowering code large language models with evol-instruct. *ArXiv*, abs/2306.08568.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2025. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235.

Rudra Murthy, Prince Kumar, Praveen Venkateswaran, and Danish Contractor. 2024. Evaluating the instruction-following abilities of language models using knowledge tasks. *arXiv preprint arXiv:2410.12972*.

Jinjie Ni, Yifan Song, Deepanway Ghosal, Bo Li, David Junhao Zhang, Xiang Yue, Fuzhao Xue, Zian Andy Zheng, Kaichen Zhang, Mahir Shah, Kabir Jain, Yang You, and Michael Shieh. 2024a. Mixeval-x: Any-to-any evaluations from real-world data mixtures. *ArXiv*, abs/2410.13754.

Jinjie Ni, Fuzhao Xue, Xiang Yue, Yuntian Deng, Mahir Shah, Kabir Jain, Graham Neubig, and Yang You. 2024b. Mixeval: Deriving wisdom of the crowd from llm benchmark mixtures. *arXiv preprint arXiv:2406.06565*.

OpenAI. 2024. Hello gpt-4o. Accessed: 2024-06-06.

Geliang Ouyang, Jingyao Chen, Zhihe Nie, Yi Gui, Yao Wan, Hongyu Zhang, and Dongping Chen. 2025. nvagent: Automated data visualization from natural language via collaborative agent workflow. *arXiv preprint arXiv:2502.05036*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *ArXiv*, abs/2304.03277.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Patrick Schober, Christa Boer, and Lothar A Schwarte. 2018. Correlation coefficients: appropriate use and interpretation. *Anesthesia & analgesia*, 126(5):1763–1768.

Guijin Son, Hyunwoo Ko, Hoyoung Lee, Yewon Kim, and Seunghyeok Hong. 2024. Llm-as-a-judge & reward model: What they can and cannot do. *arXiv preprint arXiv:2409.11239*.

Tian-Xiang Sun, Xiang-Yang Liu, Xi-Peng Qiu, and Xuan-Jing Huang. 2022. Paradigm shift in natural language processing. *Machine Intelligence Research*, 19(3):169–183.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024a. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*.

Jiahao Wang, Bolin Zhang, Qianlong Du, Jiajun Zhang, and Dianhui Chu. 2024b. A survey on data selection for llm instruction tuning. *arXiv preprint arXiv:2402.05123*.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. 2024c. Mixture-of-agents enhances large language model capabilities. *arXiv preprint arXiv:2406.04692*.

Peiqi Wang, Yikang Shen, Zhen Guo, Matthew Stallone, Yoon Kim, Polina Golland, and Rameswar Panda. 2024d. Diversity measurement and subset selection for instruction tuning datasets. *arXiv preprint arXiv:2402.02318*.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Want to reduce labeling cost? gpt-3 can help. *ArXiv*, abs/2108.13487.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. How far can camels go? exploring the state of instruction tuning on open resources. *ArXiv*, abs/2306.04751.

Zifeng Wang, Chun-Liang Li, Vincent Perot, Long T. Le, Jin Miao, Zizhao Zhang, Chen-Yu Lee, and Tomas Pfister. 2024e. Codeclm: Aligning language models with tailored synthetic data. In *NAACL-HLT*.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxin Xu, et al. 2024. Benchmarking complex instruction-following with multiple constraints composition. *arXiv preprint arXiv:2407.03978*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*.

Siyuan Wu, Yue Huang, Chujie Gao, Dongping Chen, Qihui Zhang, Yao Wan, Tianyi Zhou, Xiangliang Zhang, Jianfeng Gao, Chaowei Xiao, et al. 2024a. Unigen: A unified framework for textual dataset generation using large language models. *arXiv preprint arXiv:2406.18966*.

Yang Wu, Huayi Zhang, Yizheng Jiao, Lin Ma, Xiaozhong Liu, Jinhong Yu, Dongyu Zhang, Dezhi Yu, and Wei Xu. 2024b. Rose: A reward-oriented data selection framework for llm task-specific instruction tuning. *arXiv preprint arXiv:2412.00631*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. *ArXiv*, abs/2402.04333.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *ArXiv*, abs/2304.12244.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2024a. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *ArXiv*, abs/2406.08464.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, and Radha Poovendran. 2024b. Stronger models are not stronger teachers for instruction tuning. *arXiv preprint arXiv:2411.07133*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Ke-Yang Chen, Kexin Yang, Mei Li, Min Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yunyang Wan, Yunfei Chu, Zeyu Cui, Zhenru Zhang, and Zhi-Wei Fan. 2024a. Qwen2 technical report. *ArXiv*, abs/2407.10671.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024b. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, et al. 2024. Justice or prejudice? quantifying biases in llm-as-a-judge. *arXiv preprint arXiv:2410.02736*.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in Neural Information Processing Systems*, 36:55734–55784.

Jieyu Zhang, Le Xue, Linxin Song, Jun Wang, Weikai Huang, Manli Shu, An Yan, Zixian Ma, Juan Carlos Niebles, Silvio Savarese, Caiming Xiong, Zeyuan Chen, Ranjay Krishna, and Ran Xu. 2024. Provision: Programmatically scaling vision-centric instruction data for multimodal language models. *ArXiv*, abs/2412.07012.

Wenting Zhao, Xiang Ren, John Frederick Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatgpt interaction logs in the wild. *ArXiv*, abs/2405.01470.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P Xing, et al. 2023a. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *arXiv preprint arXiv:2309.11998*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

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

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024a. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

Huichi Zhou, Zhaoyang Wang, Hongtao Wang, Dongping Chen, Wenhan Mu, and Fangyuan Zhang. 2024b. Evaluating the validity of word-level adversarial attacks with large language models. In *Annual Meeting of the Association for Computational Linguistics*.

A Detailed Related Works

Instruction Tuning Data Selection. While Large Language Models (LLMs) like GPT-4 (Achiam et al., 2023; OpenAI, 2024) and Llama-3 (Dubey et al., 2024) excel in natural language understanding and generation, their pretraining objectives often misalign with user goals for instruction-following tasks (Murthy et al., 2024; Gao et al., 2024; Wen et al., 2024). Instruction tuning (or supervised fine-tuning) addresses this gap by refining LLMs on curated datasets of prompts and responses. Recent efforts like Vicuna (Peng et al., 2023) and LIMA (Zhou et al., 2024a) demonstrate high performance with a carefully selected small dataset, highlighting the growing importance of efficient instruction tuning and paving the way for aligning models with selected samples. This involves determining which instruction-response pairs to include in the training dataset and how to sample them effectively (Albalak et al., 2024).

Three key metrics determine instruction data quality: *Difficulty*, *Quality*, and *Diversity*. *Difficulty*, focusing mainly on the question side, is considered more valuable for model learning (Liu et al., 2024b; Lee et al., 2024; Wang et al., 2024b). IFD (Li et al., 2023b) pioneered the measurement of instruction-following difficulty for specific pairs, later enhanced by utilizing GPT-2 for efficient estimation in a weak-to-strong manner (Li et al., 2024b). *Quality*, mainly addressing the response side, measures the helpfulness and safety of model responses, typically assessed using LLM evaluators (Chen et al., 2023a, 2024b; Liu et al., 2024c; Ye et al., 2024), reward models (Son et al., 2024; Lambert et al., 2024), and gradient similarity search (Xia et al., 2024). *Diversity*, spanning both instruction and response aspects, plays a crucial role in covering various instruction formats and world knowledge, primarily improving model robustness (Bukharin and Zhao, 2023; Wang et al., 2024d). Our work stands out by addressing all three key components in data selection, introducing novel approaches to measuring difficulty from multiple LLMs' responses and ultimately enhancing model performance.

Data Synthesis for Instruction Tuning. While the development of LLMs initially relied on human-curated instruction datasets for instruction tuning (Zheng et al., 2023a; Zhao et al., 2024; Lightman et al., 2023), this approach proved time-consuming and labor-intensive, particularly as the complexity and scope of target tasks increased (Demrozi et al., 2023; Wang et al., 2021). Consequently, researchers began exploring the use of frontier LLMs to generate synthetic instruction datasets, aiming to both address these scalability challenges (Ding et al., 2023; Chen et al., 2023b, 2024d) and leverage models' advanced capabilities in developing next-generation foundation models (Burns et al., 2023; Li et al., 2024b; Charikar et al., 2024). Early approaches (Xu et al., 2023; Wang et al., 2024e; Zhou et al., 2024b; Luo et al., 2023) focused on leveraging LLMs to generate synthetic instructions through a subset of human-annotated seed instructions (Chen et al., 2023a; Wang et al., 2023), and further enhanced by few-shot (Li et al., 2024a) and attribute-guided prompting (Yu et al., 2023; Wu et al., 2024a; Huang et al., 2024). A parallel line of research explored summarizing world knowledge to create more diverse synthetic datasets, aiming to maximize the coverage of different domains and task types (Cui et al., 2023; Li et al., 2024a). Recent advancements have further streamlined this process by utilizing instructions directly from pre-trained LLMs with simple prompt templates (Xu et al., 2024a; Chen et al., 2024c; Zhang et al., 2024), significantly reducing the required custom design from human effort. While existing work has primarily focused on generating extensive, diverse, and high-quality datasets—often scaling to 100,000 examples or more—this approach introduces challenges in terms of computational efficiency and training resource requirements (Li et al., 2024d; Dubois et al., 2024).

Deriving Crowded Wisdom from Multi-LLM. Single LLM's response to a question face limitations in its representation of data (particularly cutting-edge knowledge) (Lazaridou et al., 2021; Dhingra et al., 2022; Kasai et al., 2023), skills (as no single LLM is universally optimal *empirically*) (Sun et al., 2022; Liang et al., 2022; Chen et al., 2024a), and diverse perspectives (Feng et al., 2025). Previous work has demonstrated that *online* multi-LLM wisdom (also known as compositional agent frameworks (Gupta and Kembhavi, 2023)) tends to outperform single models across various domains, providing more comprehensive and reflective solution on complex downstream tasks (Wang et al., 2024c; Hong et al., 2023; Wu et al., 2023; Li et al., 2023a; Ouyang et al., 2025; Gui et al., 2025). *Offline* crowded wisdom, where data are pre-collected rather than real-time inference, also show potential in model alignment (Gallego, 2024; Rafailov et al., 2023; Meng et al., 2025) and benchmark construction (Ni et al., 2024b,b). In this paper, we pioneer the use of *offline* multi-LLM wisdom for instruction data selection by utilizing these LLMs' responses and their reward Score as *reflections* to measure instruction-response pairs' *Difficulty* and *Quality*.

B Detailed Experiment Setups

B.1 Models & Benchmarks & Datasets Introduction

Models. In our study, the synthetic instruction dataset used for data selection consists of 19 response generators across 6 model families. These families include Qwen2 (Yang et al., 2024a), Qwen2.5 (Yang et al., 2024b), LLaMA 3 (Dubey et al., 2024), LLaMA 3.1 (Dubey et al., 2024), Gemma 2 (Team et al., 2024), and Phi-3 (Abdin et al., 2024). In our experiments, we perform supervised fine-tuning on the LLaMA3.2-3B-base/instruct (Dubey et al., 2024) and Qwen-2.5-3b-base/instruct (Yang et al., 2024b) models using the selected 1K datasets. A comprehensive overview of the models used in our study is presented in Table 5.

Benchmarks. In order to evaluate the instruction-following capabilities of the models, we use two widely-used instruction-following benchmarks: MT-Bench(Zheng et al., 2023b) and Arena-Hard (Li et al., 2024c) in our study.

MT-Bench (Zheng et al., 2023b). MT-bench is a collection of open-ended questions designed to evaluate a chatbot's performance in multi-turn conversations and its ability to follow instructions—two critical factors in aligning with human preferences. It consists of 80 high-quality multi-turn questions, which are divided into 8 categories: writing, roleplay, extraction, reasoning, mathematics, coding, knowledge I (STEM), and knowledge II (humanities/social sciences). Each category contains 10 questions. This framework provides a robust tool for assessing the practical effectiveness of LLMs and their alignment with human preferences, through meticulously designed questions and evaluations conducted by human annotators.

Arena-Hard (Li et al., 2024c). Arena-Hard is a benchmark consisting 500 challenging prompts curated by BenchBuilder. It extracts high-quality prompts from crowdsourced datasets like Chatbot Arena (Zheng et al., 2023b) and WildChat-1M (Zhao et al., 2024) without human intervention.The prompts are Scored and filtered based on seven key qualities, including specificity, domain knowledge, complexity, problem-solving, creativity, technical accuracy, and real-world applicability. This ensures that the prompts are challenging and capable of distinguishing between models. Unlike static benchmarks, Arena-Hard can be continuously updated to reflect the latest advancements in LLMs, avoiding the risk of becoming obsolete or leaking test data.

Datasets. In this paper, we conduct our experiments on Magpie-100K-Generator-Zoo(Xu et al., 2024b) because it provides a sufficiently large quantity of high-quality instruction fine-tuning data. It is a subset sampled from the MagpieAir-3M (Xu et al., 2024a) dataset, a large-scale instruction dataset. Magpie-100K contains 100,000 high-quality instructions, which are categorized into several types, including information seeking, mathematics, planning, coding and debugging, advice seeking, creative writing, reasoning, data analysis, brainstorming, editing, role-playing, and more.Each instruction has responses from 19 models across 6 model families—and their reward scores form 3 reward models. The diversity of these instructions ensures that the dataset covers a wide range of scenarios and tasks, making it suitable for instruction tuning of large language models (LLMs).

B.2 Model Training Details

Table 2 demonstrates the detailed supervised fine-tuning (SFT) hyper-parameters. We perform experiments on a server with eight NVIDIA A800-SXM4-80GB GPUs, two Intel Xeon Platinum 8358P 64-Core Processor, and 1024 GB of RAM. These experiments were conducted using LLaMA-Factory(Zheng et al., 2024).

B.3 Baseline Introduction

In this section, we present five baseline methods for comparison in our study. For each baseline, we describe its implementation details and rationale for inclusion.

Length-Based Filtering (Kwon et al., 2024). The Length method filters instructions based on their token count. We use the LLaMA 3.2 3B Instruction tokenizer to compute the number of to-

Table 5: Overview of 22 models used in our study.

Model Family	Release Date	Model ID	Size
Qwen2 (Yang et al., 2024a)	Jun, 2024	Qwen2-1.5B-Instruct	1.5B
		Qwen2-7B-Instruct	7B
		Qwen2-72B-Instruct	72B
Qwen2.5 (Yang et al., 2024b)	Sept, 2024	Qwen2.5-3B	3B
		Qwen2.5-3B-Instruct	3B
		Qwen2.5-7B-Instruct	7B
		Qwen2.5-14B-Instruct	14B
		Qwen2.5-32B-Instruct	32B
		Qwen2.5-72B-Instruct	72B
Llama 3 (Dubey et al., 2024)	Apr, 2024	Llama-3-8B-Instruct	8B
		Llama-3-70B-Instruct	70B
Llama 3.1 (Dubey et al., 2024)	Jul, 2024	Llama-3.1-8B-Instruct	8B
		Llama-3.1-70B-Instruct	70B
		Llama-3.1-405B-Instruct	405B
Llama 3.2 (Dubey et al., 2024)	Jul, 2024	Llama-3.2-3B	3B
		Llama-3.2-3B-Instruct	3B
Gemma 2 (Team et al., 2024)	Jun, 2024	Gemma-2-2B-it	2B
		Gemma-2-9B-it	9B
		Gemma-2-27B-it	27B
Phi-3 (Abdin et al., 2024)	Jun, 2024	Phi-3-mini-128k-instruct	3.5B
		Phi-3-small-128k-instruct	7B
		Phi-3-medium-128k-instruct	14B

Table 6: This table includes the hyper-parameters for supervised fine-tuning.

Hyper-parameter	Value
Learning Rate	1×10^{-5}
Number of Epochs	3
Per-device Batch Size	1
Gradient Accumulation Steps	2
Optimizer	Adamw
Learning Rate Scheduler	cosine
Warmup Steps	150
Max Sequence Length	2048

kens in each instruction. Instructions that meet the predefined length criteria are selected for further processing.

Instag-Based Selection (Lu et al., 2023). The Instag method introduces instruction tagging to analyze the supervised fine-tuning process of large language models. Our implementation follows these steps:

We use DeepSeek's API to obtain the true labels of instructions. Instructions are grouped based on their assigned labels. The complexity and diversity of each group are computed. Finally, we select a subset of instructions that exhibit the most desirable characteristics.

Direct Score Filtering The Direct Score method is inspired by the work of Chen et al. (2023) (Chen et al., 2023a), which proposes a scoring mechanism for instruction selection. We implement this approach as follows:

We use the same prompt templates as the original paper. Instead of the original scoring model, we use DeepSeek for scoring, ensuring consistency with our other experimental setups. We select the top 1,000 instructions based on their scores.

Instruction Filtering by IFD The Instruction Filtering by IFD method is based on the work of Li et al. (2023) (Li et al., 2023b), which introduces self-guided data selection to improve instruction tuning. We directly use the open-source implementation from Cherry LLM and apply the following three-step process:

Train a Pre-Experienced Model to establish prior knowledge. Compute IFD (Instruction Filtering Degree) using the Pre-Experienced Model. Filter the dataset based on IFD scores to retain high-quality instructions. To evaluate the effectiveness of IFD, we implement two versions:

IFD (with pre): Uses a trained Pre-Experienced Model to compute IFD.

IFD (no pre): Computes IFD directly using the model to be trained.

Random Sampling The Random baseline selects a random subset of X instructions. Additionally, for each instruction, we randomly select one of its 19 possible responses, ensuring that instruction-response pairs are fully randomized.

C Additional Experiment Results

C.1 Dataset Size Ablation Details

Tables 7 and 8 details the training loss, evaluation loss, and scores of Llama3.2-3b-base/instruct fine-tuned on different dataset sizes when selected with the difficulty metric. The data clearly shows a rapid increase in accuracy in when increasing the dataset sizes up to 0.5k to 1k, and marginal increases afterwards. This highlights the importance of data quality over sheer quantity in instruction tuning.

C.2 CrowdSelect Performance on LoRA

Tables 9 and 10 details the performance of CrowdSelect and various baselines combined with LoRA finetuning. CrowdSelect generally outperforms the baseline dataset selection methods on LoRA. However, more instability is found in LoRA training due to its limited learning capability compared with full finetuning.

C.3 CrowdSelect Performance on Full Finetuning

Tables 11 and 12 details the performance of CrowdSelect and various baselines combined with Full finetuning.

C.4 Foundation Metric with Clustering Performance

Table 13 details the performance of our foundation metric combined with clustering strategy.

C.5 CrowdSelect Integrated Metric Performance on Different Coefficient Combinations

Tables 14, 15, and 16 details the performance of our Integrated metric performace on 9 sets of coefficients. $w = (1, 1, 2)$ stands out as stable coefficients among all other combinations.

C.6 CrowdSelect Performance on Different Finetuning Methods

Table 17 details the performance of CrowdSelect on SFT, DPO, SimPO, and ORPO(Hong et al., 2024). Data reveals consistent and stable performance our proposed metrics, while SimPO performs best on all scenarios.

C.7 CrowdSelect Performance on Different Reward Models

Table 18 details the performance of CrowdSelect on various reward models. Data reveals the importance of reward models on finetuned model performance. However, the strong points of different reward models is scattered. While the results show that reward models are crucial for effective fine-tuning, they also reveal a nuanced landscape where the strengths of different reward models are distributed across different aspects of performance. Scattered performance highlights the need for careful consideration when selecting a reward model and reflects that current Large Language reward models are of high variance. Further research into more robust reward models for Large Language models is therefore essential.

Table 7: Performance comparison of Llama-3b-instruct with different sizes of difficulty-based selected data.

Data Size	Train Loss	Eval. Loss	MT-Bench		Arena-Hard		
			Score	Avg. Tokens	Score	95% CI	Avg. Tokens
0.25k	0.418	0.951	6.850	**301**	81.9	$(-1.2, 1.5)$	275
0.5k	0.406	1.004	**6.962**	276	83.1	$(-1.0, 1.1)$	275
1k	0.407	0.942	6.887	271	82.6	$(-1.5, 1.2)$	273
2k	0.405	0.929	6.668	301	83.1	$(-1.0, 1.4)$	273
3k	0.415	0.871	6.625	304	85.1	$(-1.3, 1.3)$	**276**
4k	0.413	0.869	6.600	279	82.4	$(-1.1, 1.7)$	268
5k	0.415	0.867	6.675	295	83.3	$(-0.7, 1.4)$	272
6k	0.414	0.857	6.572	282	**84.4**	$(-1.1, 1.3)$	265
7k	0.413	0.848	6.743	286	84.1	$(-0.9, 1.2)$	266
8k	0.411	0.836	6.618	275	83.1	$(-1.1, 1.6)$	268
9k	0.411	0.822	6.681	274	83.3	$(-1.3, 1.5)$	269
10k	0.409	0.828	6.750	279	83.6	$(-0.8, 1.7)$	266

Table 8: Performance comparison of Llama-3b with different sizes of difficulty-based selected data.

Data Size	Train Loss	Eval. Loss	MT-Bench		Arena-Hard		
			Score	Avg. Tokens	Score	95% CI	Avg. Tokens
0.25k	0.567	1.138	4.731	**492**	75.0	$(-1.1, 2.1)$	289
0.5k	0.544	1.161	4.987	392	79.1	$(-1.0, 1.7)$	289
1k	0.539	1.123	5.200	325	78.1	$(-1.4, 1.5)$	289
2k	0.534	1.094	**5.337**	309	76.9	$(-1.4, 2.2)$	**290**
3k	0.537	1.046	5.237	286	80.0	$(-1.6, 1.6)$	289
4k	0.535	1.031	5.131	287	79.7	$(-1.3, 1.5)$	289
5k	0.534	1.022	4.987	271	81.5	$(-1.0, 1.5)$	289
6k	0.531	1.019	4.943	251	81.8	$(-1.3, 1.5)$	**290**
7k	0.529	1.004	4.825	218	78.5	$(-1.2, 1.7)$	289
8k	0.526	0.990	5.093	278	81.5	$(-1.1, 1.3)$	289
9k	0.519	0.982	4.893	245	**83.2**	$(-1.5, 1.2)$	289
10k	0.517	0.983	5.137	270	82.9	$(-1.0, 1.1)$	289

Table 9: Performance comparison of lora-version of Llama-3b-base/instruct and Qwen-3b-base/instruct models with different data selection strategies.

Benchmark	**Base**	**Difficulty**		**Separability**		**Stability**	
		↓	↑	↓	↑	↓	↑
				Llama3.2-3b-instruct			
MT-Bench	6.200	6.456	6.688	6.100	6.725	6.131	**6.866**
Arena-Hard	74.4	69.6	**76.8**	69.4	72.9	69.8	74.6
Arena-Hard-95%CI	(-1.0, 1.5)	(-1.8,1.4)	(-1.5,1.9)	(-2.5,1.2)	(-1.6,1.5)	(-1.7,1.7)	(-1.7,2.0)
				Llama3.2-3b-base			
MT-Bench	4.302	4.626	**4.651**	4.631	5.040	3.538	4.369
Arena-Hard	50.0	73.1	68.0	**73.8**	73.2	60.8	73.2
Arena-Hard-95%CI	(0.0,0.0)	(-1.8,1.6)	(-1.2,1.9)	(-1.2,1.8)	(-2.0,1.1)	(-1.7,1.2)	(-1.2,1.2)
				Qwen3.2-3b-instruct			
MT-Bench	**7.138**	6.906	7.068	7.025	6.937	7.018	7.037
Arena-Hard	**81.6**	77.2	79.1	80.3	78.8	76.2	78.0
Arena-Hard-95%CI	(-1.8, 1.4)	(-1.9, 1.5)	(-2.1, 1.8)	(-1.9, 1.4)	(-1.2, 1.2)	(-1.7, 1.6)	(-1.8, 1.7)
				Qwen3.2-3b			
MT-Bench	6.043	5.137	**6.612**	6.368	6.343	5.800	6.525
Arena-Hard	69.0	**76.9**	70.7	74.1	74.2	73.7	74.2
Arena-Hard-95%CI	(-2.2, 1.6)	(-2.0, 1.8)	(-1.8, 2.4)	(-1.8, 1.5)	(-2.1, 1.5)	(-2.0, 1.3)	(-1.8, 1.9)

Table 10: Performance comparison of lora-version of Llama-3b-base/instruct and Qwen-3b-base/instruct models with pre data selection strategies as baselines.

Benchmark	**Random**	**Tags**	**Direct-Score**		**Length**		**IFD**	
			↓	↑	↓	↑	**no_pre**	**pre**
				Llama3.2-3b-instruct				
MT-Bench	6.325	6.610	6.631	6.406	6.087	5.375	6.706	**6.768**
Arena-Hard	74.2	80.1	80.0	74.8	78.1	67.5	**81.2**	79.5
Arena-Hard-95%CI	(-1.7, 1.3)	(-0.7, 0.7)	(-1.4, 1.7)	(-1.1, 1.8)	(-3.4, 2.1)	(-1.4, 0.9)	(-0.8, 1.5)	(-1.6, 1.8)
				Llama3.2-3b-base				
MT-Bench	4.637	4.575	**4.962**	4.675	4.062	4.243	4.512	4.418
Arena-Hard	76.0	76.8	76.9	75.6	67.1	70.3	73.7	**77.5**
Arena-Hard-95%CI	(-2.0, 1.6)	(-1.6, 1.8)	(-1.8, 1.7)	(-1.6, 1.4)	(-2.0, 2.0)	(-2.3, 2.2)	(-1.5, 1.5)	(-1.8, 1.4)
				Qwen2.5-3b-instruct				
MT-Bench	6.950	7.125	7.131	**7.175**	7.037	7.006	6.918	6.868
Arena-Hard	78.2	83.0	77.7	81.7	75.8	76.4	78.8	**83.1**
Arena-Hard-95%CI	(-1.5, 1.8)	(-1.7, 2.1)	(-1.6, 2.0)	(-1.7, 1.9)	(-2.0, 2.0)	(-1.4, 1.7)	(-1.3, 1.2)	(-0.8, 1.0)
				Qwen2.5-3b-base				
MT-Bench	5.887	5.616	5.417	5.750	3.981	5.637	**6.427**	5.861
Arena-Hard	76.6	**83.8**	79.3	76.5	74.3	70.4	79.7	82.2
Arena-Hard-95%CI	(-1.7, 1.5)	(-1.3, 1.2)	(-1.8, 1.2)	(-2.0, 1.7)	(-1.8, 1.6)	(-1.6, 1.9)	(-1.3, 1.0)	(-1.3, 1.0)

Table 11: Performance comparison of fft-version of Llama-3b-base/instruct and Qwen-3b-base/instruct models with different data selection strategies.

Benchmark	**Base**	**Difficulty**		**Separability**		**Stablity**	
		↓	↑	↓	↑	↓	↑
				Llama3.2-3b-instruct			
MT-Bench	6.200	6.388	**6.648**	5.937	6.581	6.225	6.625
Arena-Hard	74.4	76.5	**80.5**	80.0	77.9	75.8	77.4
Arena-Hard-95%CI	(-1.0, 1.5)	(-1.6, 1.5)	(-0.9, 1.3)	(-1.3, 1.2)	(-1.5, 1.7)	(-1.3, 0.9)	(-1.5, 1.1)
				Llama3.2-3b-base			
MT-Bench	4.302	4.506	4.738	4.731	5.056	4.675	**5.088**
Arena-Hard	50.0	78.6	76.8	81.8	**83.3**	80.0	78.3
Arena-Hard-95%CI	(0.0, 0.0)	(-1.9, 2.1)	(-1.6, 1.7)	(-1.8, 1.2)	(-1.8, 1.7)	(-1.5, 1.6)	(-1.6, 2.2)
				Qwen2.5-3b-instruct			
MT-Bench	7.138	6.906	7.182	6.919	7.269	7.056	**7.294**
Arena-Hard	81.6	82.5	81.8	81.4	**83.7**	78.1	83.5
Arena-Hard-95%CI	(-1.8, 1.4)	(-1.8, 1.5)	(-1.6, 1.3)	(-1.7, 1.6)	(-1.4, 1.2)	(-1.2, 2.0)	(-1.4, 1.4)
				Qwen2.5-3b-base			
MT-Bench	6.043	6.619	6.613	6.575	**7.075**	6.763	6.681
Arena-Hard	69.0	**80.2**	73.8	76.5	74.1	74.4	76.8
Arena-Hard-95%CI	(-2.2, 1.6)	(-1.7, 1.6)	(-2.5, 1.8)	(-1.8, 1.8)	(-1.6, 2.4)	(-1.5, 1.8)	(-1.8, 1.8)

Table 12: Performance comparison of fft-version of Llama-3b-base/instruct and Qwen-3b-base/instruct models with pre data selection strategies as baselines.

Benchmark	**Random**	**Tags**	**Direct-Score**		**Length**		**IFD**	
			↓	↑	↓	↑	**no_pre**	**pre**
				Llama3.2-3b-instruct				
MT-Bench	6.356	6.393	6.068	6.050	5.612	5.781	**6.593**	6.243
Arena-Hard	74.8	**81.6**	76.9	77.6	72.9	75.0	76.8	78.4
Arena-Hard-95%CI	(-1.5, 1.6)	(-0.2, -0.2)	(-1.5, 2.0)	(-1.7, 1.9)	(-1.9, 1.9)	(-2.4, 2.0)	(-1.2, 1.6)	(-1.7, 1.5)
				Llama3.2-3b-base				
MT-Bench	4.406	**4.562**	4.131	4.400	3.393	3.893	4.281	3.962
Arena-Hard	75.3	77.3	72.7	75.8	59.4	71.8	73.9	**77.6**
Arena-Hard-95%CI	(-2.0, 1.6)	(-1.1, 1.2)	(-2.4, 1.9)	(-1.4, 1.2)	(-1.1, 1.3)	(-1.0, 1.2)	(-1.0, 1.6)	(-1.6, 1.6)
				Qwen2.5-3b-instruct				
MT-Bench	6.793	6.818	6.506	6.768	5.881	6.931	**6.962**	6.731
Arena-Hard	78.2	**82.0**	81.2	80.8	75.6	77.7	79.0	80.4
Arena-Hard-95%CI	(-1.7, 2.0)	(-2.4, 1.6)	(-1.5, 1.8)	(-2.1, 1.7)	(-1.0, 1.2)	(-1.7, 1.7)	(-1.0, 1.5)	(-1.3, 1.0)
				Qwen2.5-3b-base				
MT-Bench	6.500	6.818	6.325	**6.900**	4.925	6.591	5.798	5.825
Arena-Hard	72.9	**79.3**	75.6	76.8	71.2	72.8	76.2	74.5
Arena-Hard-95%CI	(-2.2, 1.9)	(-2.2, 1.9)	(-1.6, 2.1)	(-1.9, 1.9)	(-1.7, 1.4)	(-2.3, 1.9)	(-1.4, 1.3)	(-1.5, 1.5)

Table 13: Performance comparison of cluster-chosen-data-fft-version of Llama-3b-base/instruct and Qwen-3b-base/instruct models with different data selection strategies.

Benchmark	Base	Random	Difficulty ↓	Difficulty ↑	Separability ↓	Separability ↑	Stability ↓	Stability ↑
Llama3.2-3b-instruct								
MT-Bench	6.200	6.743	6.256	6.675	6.094	6.619	6.275	**6.913**
Arena-Hard	74.4	80.9	81.4	82.6	**84.8**	81.9	80.0	81.8
Arena-Hard-95%CI	(-1.0, 1.5)	(-1.3, 1.4)	(-1.5, 2.0)	(-1.2, 1.8)	(-1.7, 1.4)	(-1.7, 1.7)	(-2.0, 2.2)	(-1.5, 1.7)
Llama3.2-3b-base								
MT-Bench	4.302	4.869	4.825	**5.000**	4.813	4.938	4.800	4.950
Arena-Hard	50.0	79.2	80.8	79.5	80.8	**81.9**	80.6	80.9
Arena-Hard-95%CI	(0.0, 0.0)	(-0.9, 0.9)	(-1.2, 1.7)	(-1.7, 2.2)	(-2.0, 1.6)	(-1.5, 2.1)	(-1.9, 1.8)	(-2.0, 1.6)
Qwen-3b-instruct								
MT-Bench	7.138	7.006	6.988	7.150	7.238	**7.340**	7.019	7.181
Arena-Hard	81.6	82.3	82.1	**82.6**	82.5	82.3	80.3	**82.6**
Arena-Hard-95%CI	(-1.8, 1.4)	(-1.0, 0.9)	(-1.6, 1.3)	(-1.9, 1.7)	(-2.1, 1.3)	(-1.0, 1.4)	(-1.5, 1.4)	(-1.4, 2.0)
Qwen-3b-base								
MT-Bench	6.043	**7.162**	6.575	6.800	6.856	6.875	6.819	6.869
Arena-Hard	69.0	74.6	78.2	**78.5**	78.0	75.7	73.6	76.9
Arena-Hard-95%CI	(-2.2, 1.6)	(-0.7, 1.0)	(-1.9, 2.4)	(-1.6, 1.7)	(-1.7, 1.8)	(-2.2, 2.1)	(-1.8, 1.8)	(-2.1, 1.6)

Table 14: Performance comparison of fft-version of Llama-3b-instruct on different coefficient combinations for multiple metrics with clustering.

Hyperparameter	Train Loss	Eval. Loss	MT-Bench		Arena-Hard		
			Score	Avg. Tokens	Score	95% CI	Avg. Tokens
1_1_1	0.312	0.715	6.913	307	81.8	$(-0.5, 0.8)$	266
1_-1_1	0.368	0.803	6.625	292	84.2	$(-0.7, 1.0)$	269
1_1_2	0.325	0.717	**7.103**	328	**85.5**	$(-0.8, 1.1)$	271
1_1_-1	0.294	0.617	6.650	298	82.7	$(-1.5, 1.4)$	278
1_1_1.5	0.338	0.721	6.850	312	84.7	$(-1.6, 1.3)$	266
1_-1_1.5	0.391	0.795	6.781	286	83.0	$(-1.4, 1.4)$	270
-1_-1_1	0.354	0.707	6.781	308	81.9	$(-1.5, 1.3)$	275
-1_-1_2	0.355	0.742	6.838	297	84.8	$(-1.3, 1.2)$	275
-1_-1_1.5	0.351	0.754	6.638	289	81.8	$(-1.3, 1.3)$	276

Table 15: Performance comparison of fft-version of Qwen-3b-instruct with different coefficient combinations for multiple metrics.

Hyperparameter	Train Loss	Eval. Loss	MT-Bench		Arena-Hard		
			Score	Avg. Tokens	Score	95% CI	Avg. Tokens
1_1_1	0.354	0.776	6.856	359	83.6	$(-1.7, 1.2)$	259
1_-1_1	0.432	0.861	7.138	383	81.6	$(-1.4, 1.5)$	259
1_1_2	0.371	0.776	7.131	366	**85.2**	$(-1.2, 1.1)$	262
1_1_-1	0.310	0.645	7.231	376	82.3	$(-1.6, 1.5)$	261
1_1_1.5	0.369	0.755	6.981	387	83.6	$(-2.0, 1.2)$	260
1_-1_1.5	0.430	0.872	**7.371**	390	82.4	$(-1.7, 1.5)$	260
-1_-1_1	0.431	0.874	7.025	397	81.9	$(-1.1, 1.9)$	260
-1_-1_2	0.431	0.888	6.963	377	80.6	$(-1.8, 1.5)$	259
-1_-1_1.5	0.433	0.869	6.956	377	82.4	$(-1.8, 1.3)$	260

Table 16: Performance comparison of fft-version of Llama-3b with different coefficient combinations for multiple metrics.

Hyperparameter	Train Loss	Eval. Loss	MT-Bench		Arena-Hard		
			Score	Avg. Tokens	Score	95% CI	Avg. Tokens
1_1_1	0.437	0.901	4.800	306	80.8	$(-1.3, 1.6)$	289
1_-1_1	0.497	1.007	5.019	319	80.3	$(-2.2, 2.1)$	290
1_1_2	0.454	0.904	4.613	282	82.1	$(-1.8, 1.8)$	290
1_1_-1	0.416	0.786	4.669	283	**83.0**	$(-1.6, 2.0)$	289
1_1_1.5	0.449	0.908	4.731	276	75.7	$(-1.9, 2.4)$	290
1_-1_1.5	0.496	1.016	**5.125**	309	80.6	$(-2.4, 1.6)$	290
-1_-1_1	0.469	0.973	5.050	307	80.7	$(-1.8, 1.2)$	289
-1_-1_2	0.469	0.968	4.719	268	81.6	$(-1.2, 1.1)$	290
-1_-1_1.5	0.469	0.968	4.588	291	80.0	$(-2.0, 1.8)$	290

Table 17: Performance comparison of Llama-3b-instruct models with different finetuning methods

Benchmark	Random	Difficulty ↓	Difficulty ↑	Separability ↓	Separability ↑	Stability ↓	Stability ↑
SFT							
MT-Bench	6.200	6.388	**6.648**	5.937	6.581	6.225	6.625
Arena-Hard	74.4	76.5	**80.5**	77.9	80.0	75.8	77.4
Arena-Hard-95%CI	(-1.0, 1.5)	(-1.6, 1.5)	(-0.9, 1.3)	(-1.5, 1.7)	(-1.3, 1.2)	(-1.3, 0.9)	(-1.5, 1.1)
DPO							
MT-Bench	6.463	6.431	6.768	6.431	6.418	6.256	**6.818**
Arena-Hard	74.2	75.1	77.3	76.1	**78.5**	73.2	76.2
Arena-Hard-95%CI	(-1.8, 1.6)	(-1.6, 1.6)	(-1.6, 1.7)	(-1.9, 1.9)	(-1.5, 1.4)	(-1.4, 1.3)	(-1.9, 1.5)
SimPO							
MT-Bench	6.950	6.425	**7.137**	6.518	7.043	6.675	6.931
Arena-Hard	78.7	78.0	78.8	78.2	**79.7**	76.0	75.5
Arena-Hard-95%CI	(-2.5, 2.0)	(-2.5, 3.1)	(-0.9, 1.2)	(-1.6, 0.8)	(-5.4, 6.5)	(-1.3, 1.1)	(-5.7, 6.2)
ORPO							
MT-Bench	6.412	6.450	6.450	**6.525**	6.431	6.312	6.400
Arena-Hard	73.7	73.2	73.7	73.3	74.6	73.2	**75.6**
Arena-Hard-95%CI	(-2.1, 2.2)	(-2.2, 1.8)	(-1.5, 2.0)	(-1.9, 1.8)	(-2.0, 2.2)	(-2.1, 2.2)	(-1.8, 2.2)

Table 18: Performance comparison of lora-version of Llama-3b-instruct models with different reward-models

Benchmark	**Difficulty**		**Separability**		**Stability**		**Reward-Score**	
	↓	↑	↓	↑	↓	↑	↓	↑
ArmoRM-Llama3-8B-v0.1								
MT-Bench	6.625	**6.687**	6.468	6.493	6.375	6.431	4.037	6.512
Arena-Hard	**81.7**	78.6	74.3	75.6	77.3	80.0	57.8	83.2
Arena-Hard-95%CI	(-2.0, 1.8)	(-1.8, 1.8)	(-1.8, 2.1)	(-2.0, 1.6)	(-1.8, 2.0)	(-1.0, 1.8)	(-2.0, 1.9)	(-1.5, 1.9)
Skywork-Reward-Llama-3.1-8B								
MT-Bench	6.456	6.688	6.100	6.725	6.131	**6.866**	4.012	6.675
Arena-Hard	69.6	76.8	69.4	72.9	69.8	74.6	52.6	**77.4**
Arena-Hard-95%CI	(-1.5,1.9)	(-1.8,1.4)	(-2.5,1.2)	(-1.6,1.5)	(-1.7,1.7)	(-1.7,2.0)	(-2.4, 2.0)	(-1.8, 2.1)
Skywork-Reward-Gemma-2-27B								
MT-Bench	6.512	6.593	6.756	6.881	6.637	6.756	3.793	**6.943**
Arena-Hard	76.2	78.2	75.4	80.2	79.7	**83.6**	56.1	79.6
Arena-Hard-95%CI	(-1.6, 2.0)	(-1.6, 1.5)	(-2.1, 2.1)	(-1.7, 2.4)	(-1.4, 1.4)	(-1.9, 2.0)	(-2.1, 2.1)	(-1.6, 1.7)

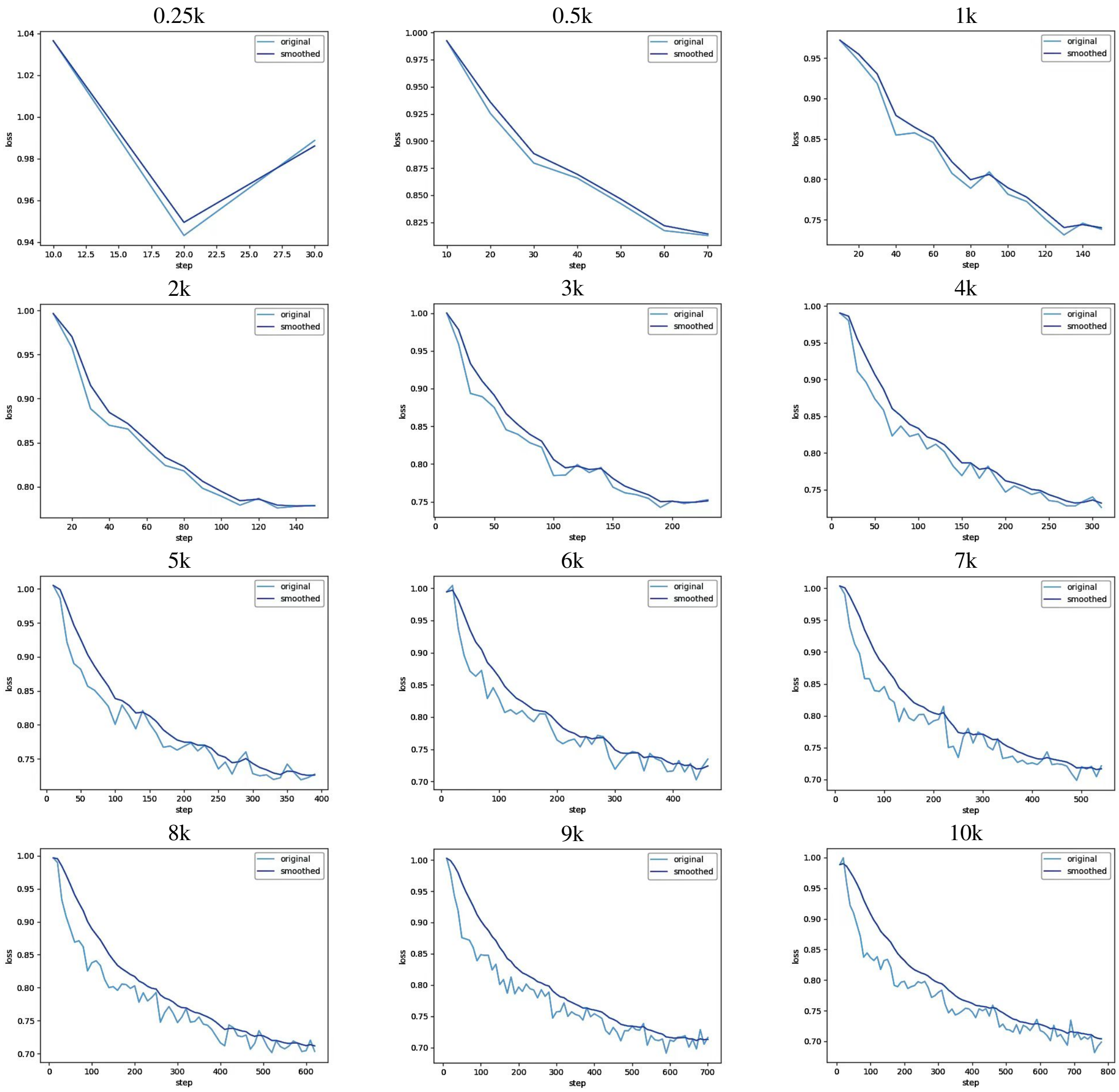

Figure 5: Lora train loss of training Llama-3b by using different sizes of randomly chosen data.