Stronger Models are Not Always Stronger Teachers for Instruction Tuning

Zhangchen Xu Fengqing Jiang Luyao Niu Bill Yuchen Lin Radha Poovendran
University of Washington
{zxu9,fqjiang,luyaoniu,byuchen,rp3}@uw.edu

Abstract

Instruction tuning has been widely adopted to ensure large language models (LLMs) follow user instructions effectively. The resulting instruction-following capabilities of LLMs heavily rely on the instruction datasets used for tuning. Recently, synthetic instruction datasets have emerged as an economically viable solution to provide LLMs diverse and high-quality instructions. However, existing approaches typically assume that larger or stronger models are stronger teachers for instruction tuning, and hence simply adopt these models as response generators to the synthetic instructions. In this paper, we challenge this commonly-adopted assumption. Our extensive experiments across five base models and twenty response generators reveal that larger and stronger models are not necessarily stronger teachers of smaller models. We refer to this phenomenon as the *Larger Models' Paradox*. We observe that existing metrics cannot precisely predict the effectiveness of response generators since they ignore the compatibility between teachers and base models being fine-tuned. We thus develop a novel metric, named as Compatibility-Adjusted Reward (CAR) to measure the effectiveness of response generators. Our experiments across five base models demonstrate that CAR outperforms almost all baselines.

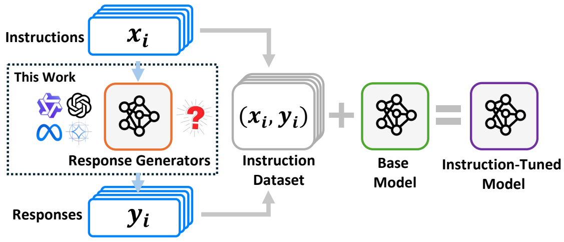

Figure 1: This figure demonstrates the process of instruction tuning and the scope of this paper.

1 Introduction

Instruction tuning (Figure 1) has been widely adopted to tailor the behavior of base Large Language Models (LLMs) to align with specific tasks and user intents (Zhang et al., 2023). This approach leverages instruction datasets, consisting of samples pairing an instruction with a corresponding response. The success of instruction tuning depends on the availability of high-quality instruction datasets. Initially, constructing these datasets required large human effort in generating and curating instruction-response pairs (Databricks, 2023; Zheng et al., 2024; Zhao et al., 2024), which is time-consuming and labor-intensive (Liu et al., 2024b).

To reduce the reliance on human-curated datasets, synthetic datasets generated by LLMs have surfaced as a viable solution (Adler et al., 2024). Recent works, such as (Sun et al., 2023; Taori et al., 2023; Wang et al., 2023; Xu et al., 2024; Chen et al., 2024), have shown the strong potential of synthetic datasets in instruction tuning. While current research has primarily focused on using LLMs to create large, diverse, and high-quality instructions (Liu et al., 2024b), the selection of appropriate LLMs for generating corresponding responses remains largely unexplored. The common approach relies on distilling from state-of-the-art models that excel in benchmark evaluations (Fourrier et al., 2024; Chiang et al., 2024) to generate responses for instruction tuning. For instance, *Llama-3.2-3B-Instruct* uses responses generated by *Llama-3.1-405B-Instruct* (i.e., the largest model in Llama-3.1 family) for instruction tuning (Meta, 2024b). Additionally, most of the existing open synthetic datasets (Teknium, 2023; Xu et al., 2023a; Ding et al., 2023; Gallego, 2023; Chen et al., 2024) depend on expensive, closed-source models like *GPT-4* (Achiam et al., 2023) and *Gemini* (Google, 2024) to produce responses.

Is it always better to use the larger or stronger models as teachers? In this paper, we investigate the choice of the teacher model that gener-

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 4392–4405
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

ate responses during synthetic dataset generation, which we refer to as **response generators**, influence the instruction-following performance of the instruction-tuned LLMs. Specifically, given a base model and a set of high-quality instructions, we investigate the following research questions:

RQ1: *Which models are the most effective response generators for instruction tuning?*

To answer RQ1, we conduct extensive experiments with five base models, and fine-tune them on datasets generated by 20 response generators across seven model families: Qwen2, Qwen2.5, Llama 3, Llama 3.1, Gemma 2, Phi-3, and GPT-4. Our findings challenge common assumptions in the field, revealing a surprising result which we term the **Larger Models' Paradox**: larger response generators (e.g., *Llama-3.1-405B-Instruct*) do not always enhance a base model's instruction-following capabilities compared to their smaller counterparts within the same model family (e.g. *Llama-3.1-70B-Instruct*). Moreover, we find that open-source models (e.g., *Gemma-2-9b-it* and *Qwen2.5-72B-Instruct*) outperform *GPT-4* as response generators. These findings question established practices and suggest more efficient and accessible approaches to create high-quality instruction datasets.

To further explore the Larger Models' Paradox, we investigate statistical metrics to reveal potential factors influencing the effectiveness of response generators. Here, we pose our second research question:

RQ2: *How can we determine the most effective response generators for a certain base model without instruction tuning?*

This question is crucial due to the significant computational costs associated with instruction tuning across multiple datasets generated by diverse response generators. Our investigation reveals that existing metrics in alignment data selection, including quality (Dubey et al., 2024), difficulty (Li et al., 2024d), and response length (Liu et al., 2023), fail to consider the **compatibility** between the base model being fine-tuned and the response generator, thus results in their inability to explain the Larger Models' Paradox. To bridge this gap, we formulate the task of finding the most effective response generators as a risk-return problem. We solve this by calculating an **Compatibility-Adjusted Reward (CAR)**, where compatibility serves as the risk factor. This compatibility is quantified by the average loss of responses on the base model being fine-tuned, with higher average loss indicating lower compatibility and thus higher risk. Our comparison of the proposed CAR with existing metrics demonstrates that it outperforms all baselines in predicting the effectiveness of response generators.

We believe that our findings on the Larger Models' Paradox and the proposed CAR can effectively guide future instruction tuning of LLMs. Instead of selecting response generators solely based on benchmark performance (e.g., GPT-4), practitioners should prioritize those with higher compatibility to better enhance the instruction-following capabilities of their LLMs.

2 Related Work

Synthetic Data Generation for Instruction Tuning. While human-crafted instruction datasets (Databricks, 2023; Zheng et al., 2024; Zhao et al., 2024) have been used for LLM instruction tuning, they are time-consuming and labor-intensive. Consequently, synthetic dataset generation has emerged as a promising alternative. Early approaches (Wang et al., 2023; Taori et al., 2023; Xu et al., 2023a,b; Wang et al., 2024b; Luo et al., 2023; Sun et al., 2023) focused on prompting LLMs to generate synthetic instructions, starting with a small set of human-annotated seed instructions and expanding these through few-shot prompting (Li et al., 2024a). Another line of work (Ding et al., 2023; Li et al., 2024a) summarized world knowledge to generate more diverse synthetic datasets. Recent advancements (Xu et al., 2024; Chen et al., 2024) further simplified the process by leveraging single prompts to sample instructions directly from LLMs, requiring minimal human oversight. While existing work primarily focused on generating large, diverse, and high-quality instructions, the impact of response generators is often overlooked.

Metrics for Data Selection. Instruction tuning data selection involves determining which instruction-response pairs to be included in the training dataset and how to sample them (Albalak et al., 2024). The most widely-used metric for selecting instruction data is quality, which is often assessed using LLM evaluators (Chen et al., 2023; Liu et al., 2024a), reward models (Dubey et al., 2024; Xu et al., 2024), gradient similarity search (Xia et al., 2024a), or a combination of these methods (Cao et al., 2024). Another key metric is difficulty, where higher difficulty is considered more valuable for learning. For instance, Li et al. (2024d) introduces IFD, which measures the instruction-following difficulty of specific instruction-response pairs. Li et al. (2024c) further refines IFD by utilizing GPT-2 for efficient estimation. Approaches like Deita (Liu et al., 2023) consider both quality and difficulty when selecting datasets. Token length is also adopted as a metric, as discussed in (Xia et al., 2024b; Liu et al., 2023). Selective Reflection-Tuning Li et al. (2024b) approach selects and refines existing instruction-following datasets to address the inconsistency between teacher and student models.

Our investigation complements existing research on alignment data selection by shifting the focus to the response generation process itself, as illustrated in Figure 1. While prior studies have concentrated on selecting the most effective instruction-response pairs with an existing instruction dataset, we explore the crucial role that response generators play in influencing the quality of instruction tuning.

Table 1: Overview of 20 response generators used in our study.

Model Family	Release Date	Model ID	Size
Qwen2 (Yang et al., 2024)	Jun, 2024	Qwen2-1.5B-Instruct Qwen2-7B-Instruct Qwen2-72B-Instruct	1.5B 7B 72B
Qwen2.5 (Team, 2024)	Sept, 2024	Qwen2.5-3B-Instruct Qwen2.5-7B-Instruct Qwen2.5-14B-Instruct Qwen2.5-32B-Instruct Qwen2.5-72B-Instruct	3B 7B 14B 32B 72B
Llama 3 (Meta, 2024c)	Apr, 2024	Llama-3-8B-Instruct Llama-3-70B-Instruct	8B 70B
Llama 3.1 (Meta, 2024c)	Jul, 2024	Llama-3.1-8B-Instruct Llama-3.1-70B-Instruct Llama-3.1-405B-Instruct	8B 70B 405B
Gemma 2 (Team et al., 2024)	Jun, 2024	Gemma-2-2b-it Gemma-2-9b-it Gemma-2-27b-it	2B 9B 27B
Phi-3 (Abdin et al., 2024)	Jun, 2024	Phi-3-mini-128k-instruct Phi-3-small-128k-instruct Phi-3-medium-128k-instruct	3.8B 7B 14B
GPT-4 (Achiam et al., 2023)	Since Mar, 2023	GPT-4 & GPT-4 Turbo	-

3 Which Models are the most effective teachers for instruction tuning?

3.1 Preliminaries

Instruction Datasets. An instruction dataset can be represented as $\mathcal{D} = (x_i, y_i)_{i=1}^{|\mathcal{D}|}$, where each sample (x_i, y_i) consists of an instruction x_i and its corresponding response y_i. In this paper, we investigate how the response generator, denoted as $\mathcal{M}$, impacts the instruction-following capabilities of models fined-tuned with $\mathcal{D}$ with $y_i = \mathcal{M}(x_i)$.

Supervised Fine-Tuning. Supervised fine-tuning (SFT) is widely adopted to enhance instruction-following capabilities of LLMs. The SFT updates the parameters θ of a pre-trained language model to minimize the negative log-likelihood loss over the instruction dataset $\mathcal{D}$. The SFT loss can be formally expressed as:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\frac{1}{|\mathcal{D}|} \sum_{(x_i, y_i) \in \mathcal{D}} \log p_\theta(y_i | x_i). \quad (1)$$

3.2 Experimental Setup

Instruction Sets. To construct diverse and high-quality instructions, we sample from the Magpie-Air-3M dataset (Xu et al., 2024), and obtain a subset of 100K high-quality instructions, denoted as **Magpie-100K**. A detailed categorization of instruction tasks is provided in Appendix A.1. Additionally, we extracted another 100K high-quality instructions from multiple sources, including UltraFeedback (Cui et al., 2023), WildChat (Zhao et al., 2024), Lmsys-Chat-1M (Zheng et al., 2024), and Alpaca-GPT-4 (Gallego, 2023). This instruction set, denoted as **Mix-100K**, contains both human-written and synthetic instructions, ensuring a comprehensive representation of instruction types.

Response Generators. Our study considers 20 response generators across 7 model families for response generation. The model families include Qwen2 (Yang et al., 2024), Qwen2.5 (Team, 2024), Llama 3 (Meta, 2024c), Llama 3.1 (Meta, 2024c), Gemma 2 (Team et al., 2024), Phi-3 (Abdin et al., 2024), and GPT-4 (Achiam et al., 2023). A comprehensive overview of the response generators is presented in Table 1. By combining the instructions with corresponding responses generated by these teacher models, we construct instruction-response pairs for instruction-tuning. By default, we use greedy decoding to generate responses. The datasets used in our experiments can be found here[1].

Base Models. We consider five base language models from different developers of varying sizes as students, including Qwen2-1.5B (Yang et al., 2024), Gemma-2-2b (Team et al., 2024), Llama-3.2-3B (Meta, 2024a), Qwen2.5-3B, (Team, 2024) and Llama-3.1-Minitron-4B-Width-Base (Llama-3.1-Minitron-4B) (Muralidharan et al., 2024).

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

Evaluation Benchmarks. To evaluate the instruction-following capabilities of the instruction-tuned models, we use two widely-used instruction-following benchmarks: **AlpacaEval 2** (**AE2**) (Li et al., 2023) and **Arena-Hard** (**AH**) (Li et al., 2024e). Specifically, AE2 contains 805 representative instructions from real user interactions. AH contains 500 challenging user queries. AE2 and AH use *GPT-4-Turbo (1106)* and *GPT-4-0314* as the baselines to assess the performance of instruction-tuned models, respectively. Both benchmarks compare responses generated by the model of interest with those generated by baselines, and employ GPT evaluators to automatically annotate which response is preferred.

Evaluation Metrics. Similar to existing studies, we adopt two metrics to measure the performance of fine-tuned SLMs. The first metric, used by both benchmarks, is the **win rate (WR)**, which calculates the fraction of responses that are favored by the GPT evaluator. The second metric, used by AE2, is the **length-controlled win rate (LC)** (Dubois et al., 2024). LC accounts for response length to reduce its impact on WR. Additionally, we report the **Average Performance (AP)**, computed as the mean of AE2's LC and AH's WR.

Instruction-Tuning and Evaluation Setup. We use SFT and implement a cosine learning rate schedule with a max learning rate of 2×10^{-5} to fine-tuning the base models for 2 epoches (Touvron et al., 2023). The detailed hyper-parameters and experimental platform can be found in Appendix A.2. We follow the official instruction templates of each model. To ensure reproducibility of our empirical analysis, we implement greedy decoding for both AE2 and AH benchmarks.

3.3 Empirical Evaluation

This section evaluates the instruction-following capabilities of models fine-tuned over datasets whose responses are generated by various response generators. By default, we utilize the Magpie-100K dataset as our primary instruction set. Figure 2 provides a comprehensive overview of the AP across different base models and response generators, and the detailed benchmark scores of AE2 and AH are deferred to Table 7 in Appendix B.1. Evaluations on larger base model (Llama-3.1-8B) with different response generators are presented in Table 6 in Appendix B.2. We analyze the effect of data randomness on average performance in Table 8.

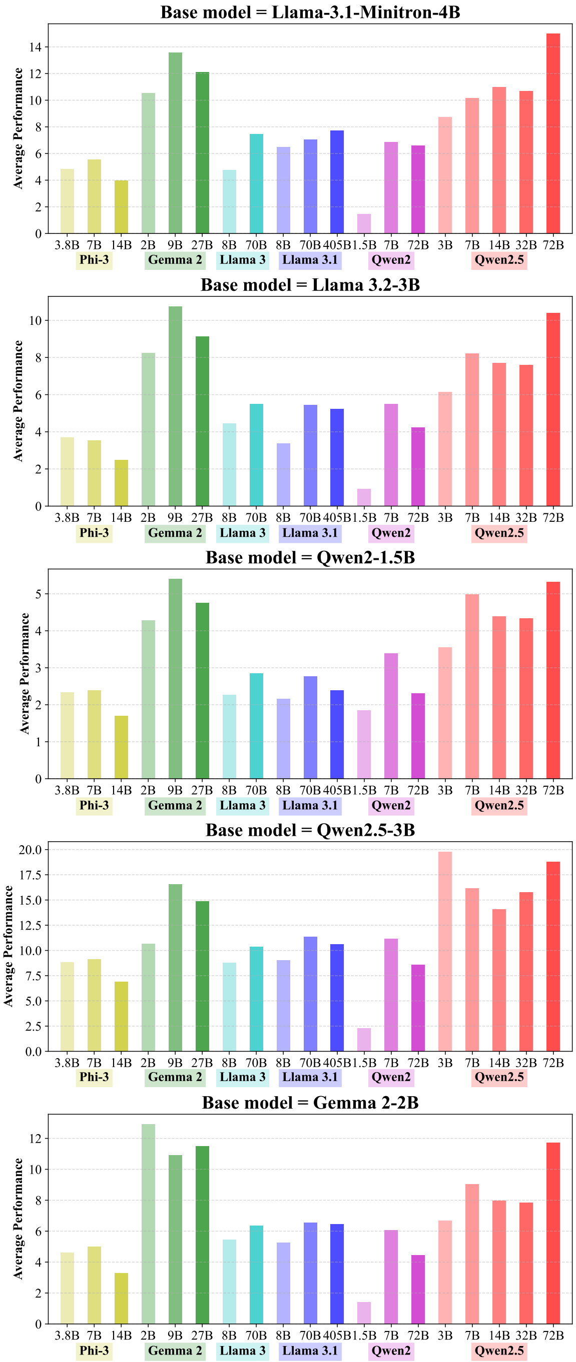

Figure 2: Average performance of five base models fine-tuned on various response generators across six model families. We use different colors to distinguish between model families, with darker bars indicating larger response generators within each family.

We observe that the Gemma-2 and Qwen2 families consistently demonstrate superior performance across all base models evaluated. Notably, **Gemma-2-9b-it** and **Qwen2.5-72B-Instruct** emerge as the two best response generators, as evidenced by their consistently high AP scores. In addition, we report the following key findings.

Finding 1: [Larger Models' Paradox] Larger response generators $\not\Rightarrow$ improved instruction-following capabilities.

Our evaluation reveals a *counterintuitive finding*: increasing the model size of response generators does not necessarily improve the instruction-following capabilities of base models within the same model family. This finding is universal, evidenced across multiple model families. For example, *Gemma-2-9b-it* demonstrates superior performance compared to its larger counterpart, *Gemma-2-27b-it*, in SFT across almost all base models examined. Similar observations are made in other model pairs: *Phi-3-Small* outperforms *Phi-3-Medium*, *Llama-3.1-70B-Instruct* surpasses *Llama-3.1-405B-Instruct*, *Qwen2-7B-Instruct* outperforms *Qwen2-72B-Instruct*, and *Qwen2.5-7B-Instruct* exceeds *Qwen2.5-32B-Instruct*. We refer to this finding as the **Larger Models Paradox**: larger language models, despite their superior performance, may not always generate better responses for fine-tuning smaller language models within the same model family compared to responses generated by medium-sized models.

We believe the key to explain this paradox is the **compatibility** between the response generators and base models. For example, a high-quality textbook (responses from large size response generators) written for college students may be challenging for primary school students (smaller base models). We will investigate this paradox in Section 4 with more detailed statistics and metrics to evaluate the compatibility.

Finding 2: [Family's Help] Learning from response generators within the same model family leads to higher performance.

We observe higher AP when base models are fine-tuned using responses generated by models within the same family. This is evidenced when Qwen2-1.5B, Qwen2.5-3B, and Gemma 2-2B serve as base models. In these instances, the relative performance of using intra-family response generators surpasses that observed when tuning other base models.

Furthermore, while not practically applicable, we observe a significant performance boost when fine-tuning a base model using responses generated from its own instruction-tuned version. A prime example of this is the Gemma 2-2B base model, which achieves best performance when tuned with responses from *Gemma-2-2b-it*, outperforming all other response generators. These two phenomena underscore the importance of compatibility between the base model and the response generator in instruction tuning.

Table 2: This table compares the performance of GPT-4 and other state-of-the-art open source LLMs as the response generator. All models are supervised-fine-tuned on the Llama-3.1-Minitron-4B base model.

Response Generator Model	AlpacaEval 2 LC (%)	AlpacaEval 2 WR (%)	Arena-Hard WR (%)	AP (%)
Gemma-2-9b-it	16.09	13.70	13.7	14.90
Gemma-2-27b-it	13.93	13.31	12.4	13.17
Llama-3-70b-Instruct	10.55	10.68	6.7	8.62
Llama-3.1-70b-Instruct	9.52	10.10	8.3	8.91
Qwen2.5-7B-Instruct	13.50	14.33	10.6	12.05
Qwen2.5-72B-Instruct	**19.20**	**21.01**	**13.1**	**16.15**
GPT-4	6.63	5.70	4.8	5.72

Finding 3: [Open-Source > Close-Source] Open-source LLMs can outperform close-source LLMs as response generators.

Table 2 compares the instruction-tuning performance when utilizing GPT-4 and open-source LLMs (e.g., Gemma 2, Llama 3, Llama 3.1 and Qwen2.5) as response generators. For this evaluation, we employ the Mix-100K dataset as our instruction source. Notably, our findings reveal that all open-source LLMs significantly outperform GPT-4. We hypothesize that this is because the response length of GPT-4 is less than open-source LLMs, thus less favored by the evaluators. These results suggest the potential for using cost-effective open-source LLMs for synthetic data generation in instruction-tuning tasks.

Finding 4: Higher temperature and top-p enhance instruction-following capabilities.

Figure 3 illustrates the effects of different sampling hyper-parameters when generating responses using *Gemma-2-9b-it* model. We observe that higher temperature and top-p value can lead to better performance in instruction following. We hypothesize that this enhancement in performance is because higher temperature and top-p values yield more diverse and contextually rich outputs.

Finding 5: Reject sampling slightly increases instruction-tuning performance.

Table 3 quantifies the impact of reject sampling

Table 3: This table investigates the impact of reject sampling on model performance.

Base Model	Method	AlpacaEval 2 LC (%)	AlpacaEval 2 WR (%)	Arena-Hard WR (%)	AP (%)
Llama-3.1-Minitron-4B	Best-of-N	**15.94**	**15.14**	**11.9**	**13.92**
	Worst-of-N	13.02	12.66	11.0	12.01
	Sampling	15.71	14.81	11.8	13.755
	Greedy	16.13	14.51	11.0	13.565
Qwen2.5-3B-Instruct	Best-of-N	**13.83**	**13.57**	**21.0**	**17.415**
	Worst-of-N	12.37	12.54	17.9	15.135
	Sampling	13.43	13.29	20.1	16.765
	Greedy	13.78	13.57	19.4	16.59

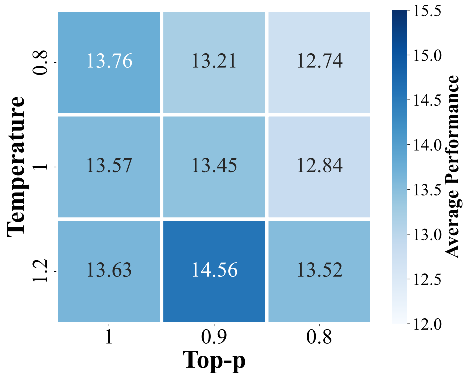

Figure 3: This figure demonstrates the impact of different sampling hyper-parameters when generating responses. We use *Gemma-2-9b-it* as the response generator. All models are supervised-fine-tuned on the Llama-3.1-Minitron-4B base model.

on synthetic data generation using *Gemma-2-9b-it* model. Specifically, we generate 5 responses per instruction with temperature $T = 0.8$, evaluate them using the *ArmoRM-Llama3-8B-v0.1* reward model (Wang et al., 2024a), and select the highest and lowest-rated responses to create two distinct datasets: Best-of-N and Worst-of-N. We also compare them with responses sampled at $T = 0.8$ and greedy decoding ($T = 0$). The results presented in Table 3 demonstrate a slight improvement in performance when utilizing reject sampling compared to standard sampling techniques.

In what follows, we summarize the conclusion for RQ1.

RQ1. Which models are the most effective response generators for instruction tuning?

A1. Gemma-2 and Qwen2 families consistently demonstrate superior performance across all base models evaluated, and even **outperform GPT-4**. Notably, **Gemma-2-9b-it** and **Qwen2.5-72B-Instruct** emerge as the two best response generators, as evidenced by their consistently high AP scores. We also found that **larger models do not always generate responses for enhanced instruction-following capabilities**.

4 How can we determine the most effective response generators without instruction tuning?

4.1 Measure the Effectiveness of Response Generators

It is computationally expensive to brute-force all response generators to identify the most effective one for a given base model. In this section, we investigate how to measure the effectiveness of response generators for a given base model without training or fine-tuning. Specifically, we study the following research question:

Definition 4.1 (Effectiveness Measure of Response Generators)**.** Given a base language model and a set of synthetic instruction datasets $\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_n$, where each $\mathcal{D}_i$ contains responses generated by a distinct response generator $\mathcal{M}_i$, measure the effectiveness of these response generators without performing the actual fine-tuning process.

Evaluation Metric. To assess the accuracy when measuring effectiveness of response generators, we employ Spearman's rank correlation coefficient (ρ) (Zar, 2005). This coefficient evaluates the monotonic relationship between two ranking variables. In our context, we compute ρ between two ranks: the ground truth rank R_{AP}, obtained by fine-tuning the model on each synthetic instruction dataset and measuring the Average Performance (AP), and an estimated rank R_{EST}, predicted without fine-tuning. Spearman's ρ is calculated as:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{2}$$

where d_i is the difference between the two ranks for each observation and n is the number of observations. ρ ranges from -1 to 1, with 1 indicating a perfect positive correlation. Our objective is to maximize ρ, thereby achieving the closest prediction between predicted and actual performance rankings. We employ the empirical results obtained in Section 3 as the ground truth.

4.2 Baseline Methods

In this section, we introduce commonly-used metrics for alignment data selection: quality, difficulty, and response length, for predicting the performance rank of instruction-tuned models.

Response Quality. Following Meta (2024a); Xu et al. (2024), we assess response quality using reward models and calculate the **Average Reward (AR)** of all responses. To mitigate potential selection bias, we employ 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).

Instruction-following Difficulty. Instruction-following difficulty is another widely-used metric in alignment data selection (Meta, 2024a; Liu et al., 2023; Li et al., 2024d,c; Xu et al., 2024). To assess the difficulty of responses, we employ the following two metrics:

1. **Response Perplexity (PPL).** For a given instruction-response pair (x_i, y_i), the response perplexity is defined as:

$$\text{PPL}(y_i|x_i) = \exp\left(-\frac{1}{N}\sum_{j=1}^{N}\log p_\theta(y_{i,j}|x_i, y_{i,1:j-1})\right),$$

where N is the token length of y_i and $y_{i,j}$ is its j-th token, and θ is the parameter of the base model. We use *GPT-2* model and each corresponding base model for evaluation, denoted as PPL-GPT2 and PPL-Self respectively.

2. **Instruction Following Difficulty (IFD) (Li et al., 2024d).** IFD is defined as:

$$\text{IFD}(y_i|x_i) = \frac{\text{PPL}(y_i|x_i)}{\text{PPL}(y_i)},$$

where $\text{PPL}(y_i)$ is the unconditional perplexity of response y_i. We follow Li et al. (2024c) and employ *GPT-2* and the base model respectively, denoted as IFD-GPT2 and IFD-Self.

For each metric, we compute the average value across the entire dataset $\mathcal{D}_i$.

Response Length. According to Liu et al. (2023) and Xia et al. (2024b), the response length positively correlates with the final alignment performance. We use the `tiktoken` library (OpenAI, 2024) to count the number of response tokens for each pair, and report the average response length for each $\mathcal{D}_i$.

4.3 Baseline Methods Fails to Measure the Effectiveness of Response Generators

In what follows, we demonstrate that the effectiveness of response generators indicated by baseline methods does not match the performance of models fine-tuned on various synthetic instruction datasets.

As shown in Figure 4, AR consistently increases with model size within model families (except Phi-3 family). However, this trend fails to explain the "Larger Models Paradox" discussed in Section 3. Notably, since AR measures human preference, this discrepancy suggests that responses preferred by humans are not necessarily optimal for aligning language models.

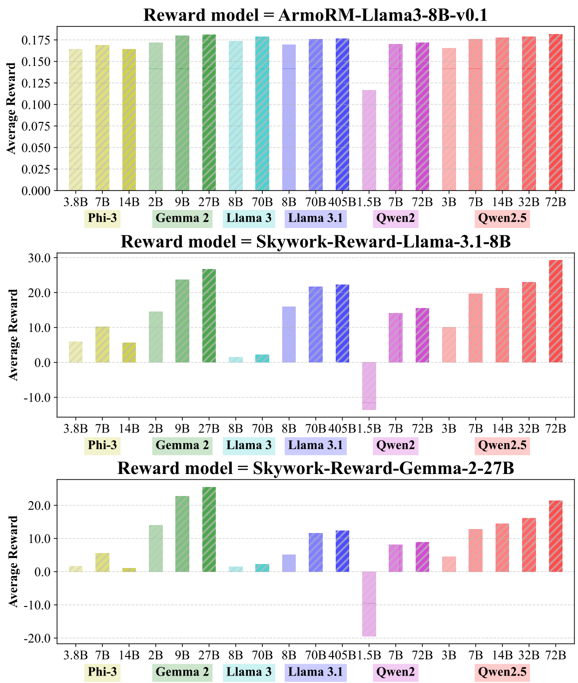

Figure 4: This figures demonstrates the response quality measured by three reward models.

Similarly, metrics representing instruction-following difficulty (IFD and Perplexity) and response length show no strong correlation with model instruction-following capabilities. We deferred the results and analysis of these metrics to Appendix B.4. These findings highlight the inadequacy of existing metrics in accurately measuring

Table 4: Spearman's rank correlation coefficient (ρ) for different measurement metrics. Here $\mathcal{RM}_1$, $\mathcal{RM}_2$, $\mathcal{RM}_3$ are reward models *ArmoRM-Llama3-8B-v0.1*, *Skywork-Reward-Llama-3.1-8B*, and *Skywork-Reward-Gemma-2-27B* respectively. We observe that our proposed CAR shows the highest correlation between the effectiveness of the response generator and the instruction-following capabilities of fine-tuned base models.

Base Models	**Reward**			**Difficulty**				**Response Length**	**CAR**
	$\mathcal{RM}_1$	$\mathcal{RM}_2$	$\mathcal{RM}_3$	IFD-GPT2	IFD-Self	PPL-GPT2	PPL-Self		
Qwen2-1.5B	0.5526	0.7895	0.8754	0.7088	0.7719	0.1473	0.5596	0.5404	**0.8842**
Gemma 2-2B	0.5526	0.7982	0.8842	0.8281	0.8930	0.1614	0.4351	0.6298	**0.9000**
Qwen2.5-3B	0.4526	0.7351	0.7456	0.7386	0.8088	0.0456	-0.0614	0.6088	**0.8105**
Llama 3.2-3B	0.6088	0.8105	**0.9088**	0.7632	0.8579	0.0456	0.6018	0.5877	0.9053
Llama-3.1-Minitron-4B	0.6632	0.8860	0.9386	0.7491	0.8555	0.1579	0.6263	0.5807	**0.9439**
Average	0.5660	0.8039	0.8705	0.7575	0.8374	0.1116	0.4323	0.5895	**0.8888**

the effectiveness of response generators in enhancing performance of instruction-tuned models.

4.4 A Compatibility-Aware Metric to Measure Effectiveness

In this section, we present a new metric to measure the effectiveness of response generators, making the "Larger Models Paradox" explainable. Our key insight to capture the **compatibility of response generators with base models**. To reflect such compatibility, we use the loss of the response r_i in the base model being fine-tuned as the key metric. Intuitively, a lower loss of response y_i on the base model indicates that the response aligns well with the base model's existing knowledge and capabilities, thus is more learnable compared to the response with higher loss.

While compatibility is crucial, it alone cannot fully measure effectiveness. Consider a scenario where a response generator consistently produces simple, low-quality responses for every question. In such cases, although these responses might be highly compatible with the base model, their overall quality and would be low. Therefore, to bridge this gap between quality and compatibility, we formulate the task of finding the most effective response generator as a risk-return problem (Fama and MacBeth, 1973). We propose an adjusted reward value that incorporates both the potential benefit (return) and the compatibility risk. Specifically, we define our **Compatibility-Adjusted Reward (CAR)** as follows:

$$\mathrm{CAR}(\mathcal{D}_i, \theta) = \frac{r(\mathcal{D}_i)}{1 + \beta \cdot L(\mathcal{D}_i, \theta)} \quad (3)$$

where $r(\mathcal{D}_i)$ is the average reward measured by the reward model, representing the potential return, and $L(\mathcal{D}_i, \theta) = -\frac{1}{|\mathcal{D}_i|}\sum_{y_i \in \mathcal{D}_i} \log p_\theta(y_i)$ is the average loss for responses in $\mathcal{D}_i$ on the base model parameterized by θ. β is a tunable parameter that controls the impact of compatibility on the adjusted reward. CAR penalizes the average reward from the reward model with the compatibility risk measured by the loss. This balanced approach enables quantitative assessment of the trade-off between the response quality and compatibility.

4.5 Experimental Results

Table 4 compares the Spearman's ρ correlation coefficient of baseline metrics with our CAR when using datasets generated by different response generators to fine-tune various base models. For CAR calculation, we employ *Skywork-Reward-Gemma-2-27B* as the reward model and set $\beta = 3$. The results in Table 4 demonstrate that our proposed CAR consistently outperforms other baseline metrics across almost all settings, indicating its potential to predict the effectiveness of different response generators without instruction tuning.

RQ2. How can we determine the most effective response generators without instruction tuning?

A2. Existing metrics in instruction data selection are inadequate for accurate prediction as they fail to consider the compatibility between the base model and the response generator. To address this limitation, we propose the Compatibility-Adjusted Reward (CAR), which achieves better performance in identifying effective response generators across various base models.

5 Conclusion and Future Work

This paper investigates the impact of response generators in synthetic dataset generation for instruction tuning. We uncovered the Larger Models' Paradox, wherein larger response generators do not necessarily enhance a base model's instruction-following capabilities compared to their smaller counterparts within the same model family. To explain this phenomenon, we considered the compatibility between response generators and the base model, and proposed the Compatibility-Adjusted Reward (CAR). Our metric achieved better performance in identifying the effectiveness of different response generators without the need for fine-tuning, outperforming existing baselines in alignment dataset selection.

We will explore several promising directions. First, efficiently transforming existing datasets to achieve better compatibility can lead to more effective use of available instruction tuning datasets. Second, investigating theoretical foundations of compatibility would enhance our understanding of the underlying mechanisms of instruction tuning. Lastly, studying the impact of different response generators for preference tuning may help aligning LLMs to better reflect human values.

Limitations

While our study provides valuable insights into the effectiveness of response generators in instruction tuning, we acknowledge that our research primarily focuses on general instruction following tasks and does not extensively explore the synthesis of alignment datasets for specialized domains such as mathematics or complex reasoning. As a result, the applicability of the Larger Models' Paradox to these specific areas remains uncertain.

Ethical Impact

This paper makes a counterintuitive observation, referred to as the Larger Models' Paradox, showing that stronger models are not stronger teachers for instruction tuning. We further propose a new metric to measure the effectiveness of teachers when generating responses for instruction datasets. This metric informs the selection of response generators for future fine-tuning processes to enhance language models' instruction-following capabilities. We do not identify potential misuse and ethical concerns in this paper.

Acknowledgment

This work is partially supported by the Air Force Office of Scientific Research (AFOSR) under grant FA9550-23-1-0208, the National Science Foundation (NSF) under grants IIS 2229876, and the Office of Naval Research under grant N0014-23-1-2386.

This work is supported in part by funds provided by the National Science Foundation, by the Department of Homeland Security, and by IBM. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or its federal agency and industry partners.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, 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*.

Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, et al. 2024. Nemotron-4 340b technical report. *arXiv preprint arXiv:2406.11704*.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. 2024. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*.

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. 2024. Instruction mining: Instruction data selection for tuning large language models. In *First Conference on Language Modeling*.

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

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

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *Preprint*, arXiv:2403.04132.

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 preprint arXiv:2310.01377*.

Databricks. 2023. Databricks dolly-15k.

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 preprint arXiv: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, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Eugene F Fama and James D MacBeth. 1973. Risk, return, and equilibrium: Empirical tests. *Journal of political economy*, 81(3):607–636.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. 2024. Open llm leaderboard v2. `https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard`.

Victor Gallego. 2023. alpaca-gpt4.

Google. 2024. Our next-generation model: Gemini 1.5.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Rewardbench: Evaluating reward models for language modeling.

Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, et al. 2024a. Synthetic data (almost) from scratch: Generalized instruction tuning for language models. *arXiv preprint arXiv:2402.13064*.

Ming Li, Lichang Chen, Jiuhai Chen, Shwai He, Jiuxiang Gu, and Tianyi Zhou. 2024b. Selective reflection-tuning: Student-selected data recycling for llm instruction-tuning. *arXiv preprint arXiv:2402.10110*.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024c. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14255–14273, Bangkok, Thailand. Association for Computational Linguistics.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024d. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7602–7635, Mexico City, Mexico. Association for Computational Linguistics.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024e. From live data to high-quality benchmarks: The arena-hard pipeline.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

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

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

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, et al. 2024b. Best practices and lessons learned on synthetic data for language models. *arXiv preprint arXiv:2404.07503*.

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 preprint arXiv:2312.15685*.

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 preprint arXiv:2306.08568*.

Meta. 2024a. Llama-3.2-3b. `https://huggingface.co/meta-llama/Llama-3.2-3B`.

Meta. 2024b. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models.

Meta. 2024c. Meet llama 3.1. `https://llama.meta.com`.

Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. 2024. Compact language models via pruning and knowledge distillation. *arXiv preprint arXiv:2407.14679*.

OpenAI. 2024. Tiktoken. https://github.com/openai/tiktoken.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2023. Principle-driven self-alignment of language models from scratch with minimal human supervision. *Advances in Neural Information Processing Systems*, 36.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

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*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Teknium. 2023. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Zifeng Wang, Chun-Liang Li, Vincent Perot, Long T Le, Jin Miao, Zizhao Zhang, Chen-Yu Lee, and Tomas Pfister. 2024b. Codeclm: Aligning language models with tailored synthetic data. *arXiv preprint arXiv:2404.05875*.

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

Tingyu Xia, Bowen Yu, Kai Dang, An Yang, Yuan Wu, Yuan Tian, Yi Chang, and Junyang Lin. 2024b. Rethinking data selection at scale: Random selection is almost all you need. *Preprint*, arXiv:2410.09335.

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

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023b. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6268–6278, Singapore. Association for Computational Linguistics.

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

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, Keqin Chen, Kexin Yang, Mei Li, Mingfeng 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, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Jerrold H Zar. 2005. Spearman rank correlation. *Encyclopedia of Biostatistics*, 7.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2024. LMSYS-chat-1m: A large-scale real-world LLM conversation dataset. In *The Twelfth International Conference on Learning Representations*.

A More on Experimental Setups

A.1 Instruction Set Details

Figure 5 demonstrates the task category of instructions in our sampled Magpie-100K. We follow (Xu et al., 2024) and use *Llama-3-8B-Instruct* to tag the task categories. We note that this instruction set covers wide range of instructions across different task categories.

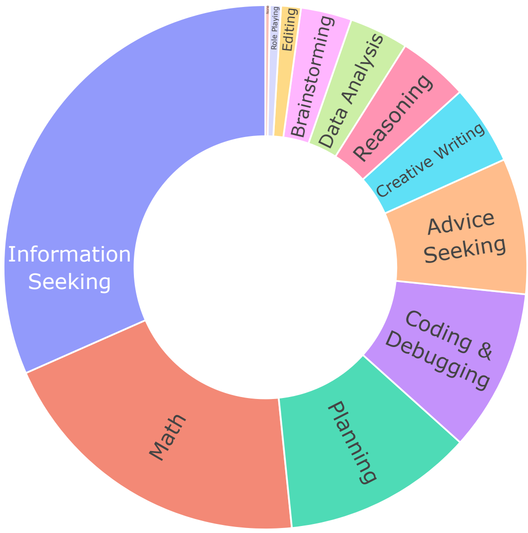

Figure 5: Task categories of the Magpie-100K instruction set used in our study.

A.2 Supervised Fine-Tuning Setups

Table 5 demonstrates the detailed supervised fine-tuning (SFT) hyper-parameters. We perform experiments on a server with four NVIDIA A100-SXM4-80GB GPUs, an AMD EPYC 7763 64-Core Processor, and 512 GB of RAM. These experiments were conducted using Axolotl[2].

Table 5: This table shows the hyper-parameters for supervised fine-tuning.

Hyper-parameter	**Value**
Learning Rate	2×10^{-5}
Number of Epochs	2
Number of Devices	4
Per-device Batch Size	1
Gradient Accumulation Steps	8
Effective Batch Size	32
Optimizer	`Adamw`
Learning Rate Scheduler	`cosine`
Warmup Steps	100
Max Sequence Length	4096

[2]https://github.com/OpenAccess-AI-Collective/axolotl

B More Experimental Results

B.1 Detailed Benchmark Scores of Instruction-Tuned LLMs

Table 7 details the benchmark scores of AE2 and AH when tuning base models with different response generators. These results complement the Average Performance shown in Figure 2.

B.2 Larger Models' Paradox in Larger Base Models

We summarize the benchmark scores of AE2 and AH when tuning large base model (Llama-3.1-8B) with diverse response generators in Table 6. We observe that the Larger Models' Paradox persists when employing the Qwen2.5 and Llama-3.1 model families as response generators. We further demonstrate that the Larger Model's Paradox is not an effect of data randomness in Table 8.

Table 6: This table presents benchmark scores of AE2 and AH when tuning large base model (Llama-3.1-8B) with diverse response generators. The Larger Models' Paradox persists when employing the Qwen2.5 and Llama-3.1 model families as response generators.

Base Model	Response Generator	AE2 LC	AE2 WR	AH	AP
Llama-3.1-8B	Qwen2.5-3B-Instruct	11.48	13.85	15.90	13.74
	Qwen2.5-7B-Instruct	18.70	20.22	25.90	21.61
	Qwen2.5-14B-Instruct	17.50	17.19	28.60	21.10
	Qwen2.5-32B-Instruct	16.20	16.42	27.80	20.14
	Qwen2.5-72B-Instruct	29.73	32.35	30.90	30.99
	Llama-3.1-8B-Instruct	12.62	14.34	15.80	14.25
	Llama-3.1-70B-Instruct	14.98	17.74	21.00	17.91
	Llama-3.1-405B-Instruct	15.40	17.00	16.50	16.30
	Gemma-2-2b-it	17.11	19.64	15.60	17.45
	Gemma-2-9b-it	25.74	22.88	23.40	24.00
	Gemma-2-27b-it	25.09	24.60	25.40	25.00

B.3 Impact of Data Randomness on Evaluation

We sample 80K instructions from Magpie-100K using different seeds and fine-tuned Llama-3.1-Minitron-4B. The performance of fine-tuned models is shown in Table 8. We observe that the average performance varies by only 2.89%, demonstrating that our evaluation is robust across different instruction samples. This finding underscores the consistency of our evaluation.

B.4 Visualization of baseline methods in measuring the effectiveness of response generators

Figure 6 presents the output length of synthetic datasets for each response generator. Figure 7 visualizes the PPL-GPT2 and IFD-GPT2 across

Table 7: This table details benchmark scores of AE2 and AH when tuning different base models with diverse response generators.

Base Model	Metric	Phi-3			Gemma 2			Llama 3		Llama 3.1			Qwen2			Qwen2.5				
		Mini	Small	Medium	2B	9B	27B	8B	70B	8B	70B	405B	1.5B	7B	72B	3B	7B	14B	32B	72B
Qwen2-1.5B	AE 2 WR	3.65	3.64	2.80	5.34	6.13	5.49	3.39	3.74	2.76	3.49	3.09	2.83	4.09	3.35	5.60	6.84	5.13	5.65	7.03
	AE 2 LC	2.85	2.98	2.18	4.16	5.60	4.99	2.64	3.10	2.10	2.74	2.36	2.68	3.47	2.82	4.50	5.66	4.38	4.96	5.83
	AH	1.8	1.8	1.2	4.4	5.2	4.5	1.9	2.6	2.2	2.8	2.4	1.0	3.3	1.8	2.6	4.3	4.4	3.7	4.8
Gemma 2-2B	AE 2 WR	6.60	6.54	4.54	16.88	11.83	12.09	7.09	8.49	7.20	9.45	8.92	2.14	7.11	6.07	7.91	12.00	8.07	9.19	16.68
	AE 2 LC	5.90	5.89	3.99	12.93	12.51	13.09	5.70	7.13	5.63	7.32	7.11	1.91	6.45	5.46	6.84	10.94	7.53	8.77	13.85
	AH	3.3	4.1	2.6	12.9	9.3	9.9	5.2	5.6	4.9	5.8	5.8	0.9	5.7	3.4	6.5	7.1	8.4	6.9	9.6
Qwen2.5-3B	AE 2 WR	8.19	7.79	5.97	10.52	13.57	10.01	8.07	10.17	7.91	9.68	9.12	2.98	8.54	6.86	16.22	12.76	10.32	11.71	18.42
	AE 2 LC	7.22	7.29	5.49	9.58	13.78	10.18	7.85	9.37	7.22	8.94	8.59	2.54	7.98	6.59	14.79	11.89	10.28	11.65	16.41
	AH	10.5	11.0	8.3	11.8	19.4	19.6	9.7	11.4	10.9	13.8	12.7	2.1	14.4	10.6	24.8	20.4	17.9	19.9	21.2
Llama-3.2-3B	AE 2 WR	4.88	3.54	3.05	8.89	11.45	10.58	4.67	5.45	4.26	6.68	6.44	1.72	6.23	5.13	6.09	7.72	6.82	7.10	12.12
	AE 2 LC	4.11	2.95	2.37	7.49	10.60	9.79	3.79	4.52	3.17	5.19	5.17	1.28	5.41	4.49	5.11	6.63	5.92	6.32	9.99
	AH	3.3	4.1	2.6	9.0	10.9	8.5	5.1	6.5	3.6	5.7	5.3	0.6	5.6	4.0	7.2	9.8	9.5	8.9	10.8
Llama-3.1-Minitron-4B	AE 2 WR	6.35	7.11	4.83	11.80	14.50	11.90	6.11	9.87	8.24	9.61	10.03	2.30	7.84	8.45	10.27	12.05	11.30	11.65	19.58
	AE 2 LC	5.74	6.61	4.31	10.37	16.13	12.34	4.80	8.93	6.96	8.52	9.23	2.03	7.31	8.11	9.17	11.12	10.89	11.13	17.77
	AH	3.9	4.5	3.6	10.7	11.0	11.9	4.7	6.0	6.0	5.6	6.2	0.9	6.4	5.1	8.3	9.2	11.1	10.2	12.2

different response generators. Figure 8 and 9 reports PPL-Self and IFD-Self, respectively. We observe that although PPL-Self and IFD-Self have higher correlation compared with measuring using GPT2, they still to fail to effectively predict the effectiveness of different response generators, with low Spearman's rank correlation coefficients demonstrated in Table 4.

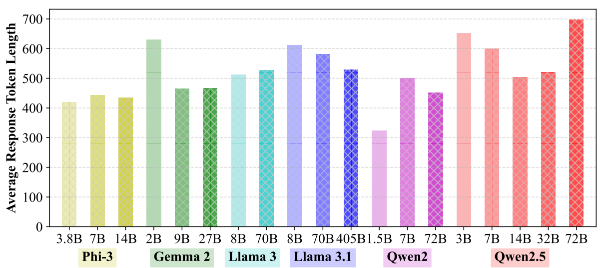

Figure 6: Average Output Length of synthetic datasets generated using different response generators (measured in Tokens).

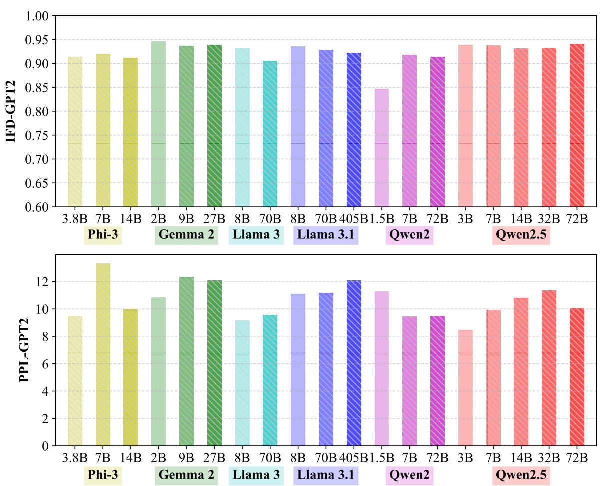

Figure 7: PPL-GPT2 and IFD-GPT2 of synthetic datasets generated using different response generators.

Table 8: We sample 80K instructions from Magpie-100K using different seeds and fine-tuned Llama-3.1-Minitron-4B with the sampled data. We observe that the average performance varies by only 2.89%, demonstrating that our evaluation is robust across different instruction samples. This finding underscores the consistency of our evaluation.

Instruction Sample	AE2 LC	AE2 WR	AH	Average Performance
Magpie-80K (Seed = 42)	14.26	13.54	12.50	13.433
Magpie-80K (Seed = 123)	13.40	12.92	12.80	13.040
Magpie-80K (Seed = 456)	14.77	12.98	11.10	12.950
Magpie-80K (Seed = 789)	13.57	12.79	11.20	12.520
Average	14.00	13.058	11.90	12.986
Standard Deviation	0.634	0.331	0.876	0.375

B.5 Impact of Reward Models on the performance of CAR

We perform ablation analysis on the choice of reward models with a weaker reward model, Skywork-Reward-Llama-3.1-8B, and calculate CAR. The Spearman's correlations are presented in Table 9. We observe that CAR using the weaker Skywork 8B reward model performs worse compared to using the stronger Skywork 27B reward model, indicating the reliance of CAR on a good performing reward model. However, even with a weaker reward model, CAR outperforms compared with using the reward model alone.

Table 9: Spearman's correlations when CAR uses different reward models. CAR relies on a good reward model. However, even with a weaker reward model, CAR outperforms compared with using the reward model alone.

Base Model	Skywork 8B	CAR (Skywork 8B)	Skywork 27B	CAR (Skywork 27B)
Qwen2-1.5B	0.7895	0.7474	0.8754	**0.8842**
Gemma 2-2B	0.7982	0.8018	0.8842	**0.9000**
Qwen2.5-3B	0.7351	0.7386	0.7456	**0.8105**
Llama-3.1-Minitron-4B	0.8860	0.8912	0.9386	**0.9439**
Llama-3.2-3B	0.8105	0.8105	**0.9088**	0.9053

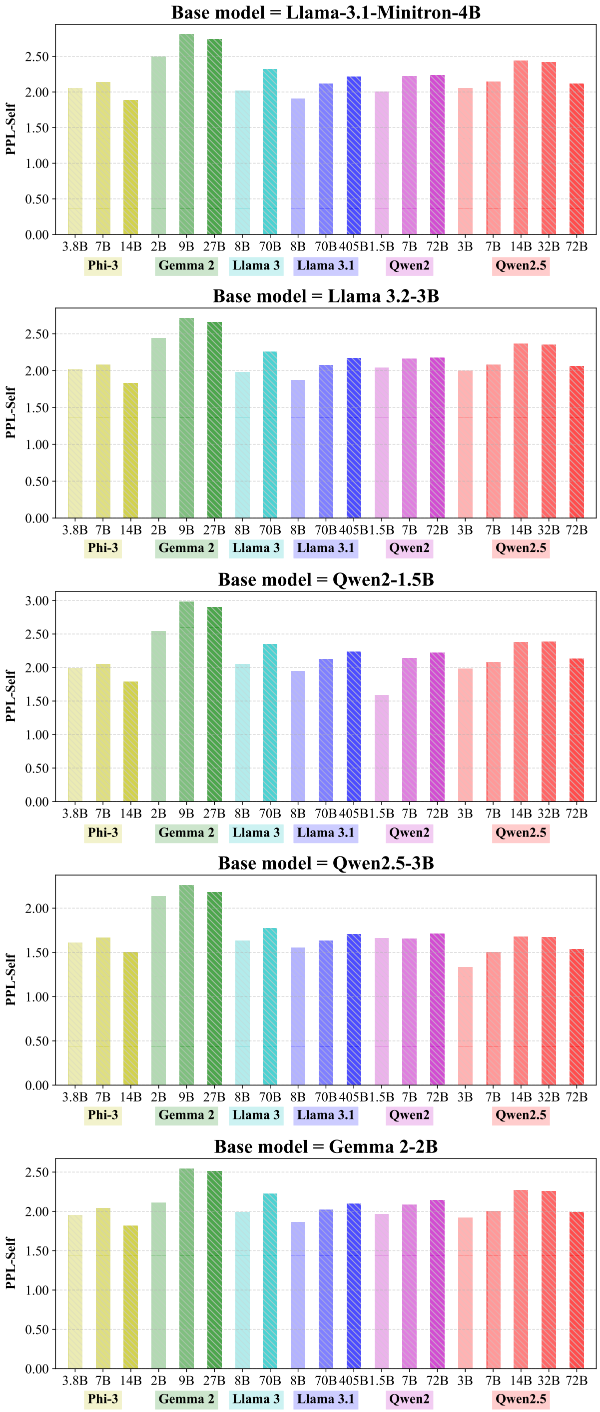

Figure 8: PPL-Self of five base models.

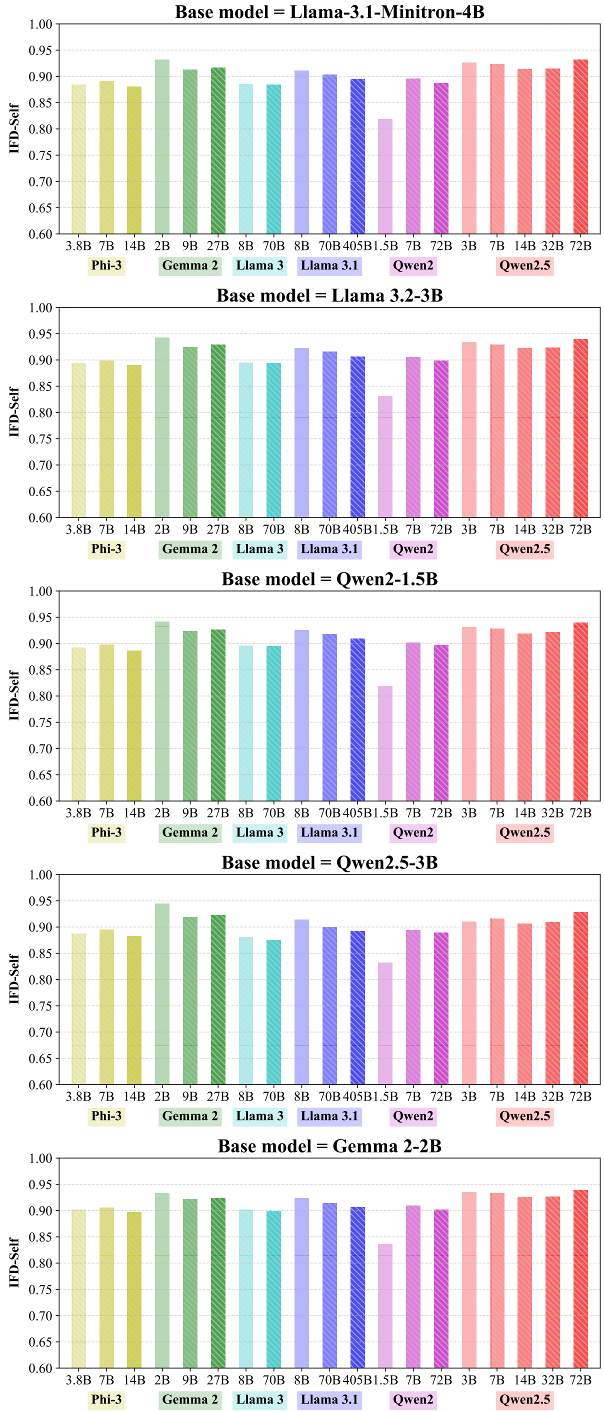

Figure 9: IFD-Self of five base models.