Importance-Aware Data Selection for Efficient LLM Instruction Tuning

Tingyu Jiang[1], Shen Li[1], Yiyao Song[1], Lan Zhang[2], Hualei Zhu[1], Yuan Zhao[1], Xiaohang Xu[*3], Kenjiro Taura[3], Hao Henry Wang[1]

[1]Alibaba Cloud Computing
[2]Independent Researcher
[3]Graduate School of Information Science and Technology, The University of Tokyo
{jiangtingyu.jty, lishen.ls, songyiyao.syy, zhuhualei.zhl, zhaoyuan.yz, qiao.wh}@alibaba-inc.com
s23221054004@smail.cczu.edu.cn
{xhxu, tau}@eidos.ic.i.u-tokyo.ac.jp

Abstract

Instruction tuning plays a critical role in enhancing the performance and efficiency of Large Language Models (LLMs). Its success depends not only on the quality of the instruction data but also on the inherent capabilities of the LLM itself. Some studies suggest that even a small amount of high-quality data can achieve instruction fine-tuning results that are on par with, or even exceed, those from using a full-scale dataset. However, rather than focusing solely on calculating data quality scores to evaluate instruction data, there is a growing need to select high-quality data that maximally enhances the performance of instruction tuning for a given LLM. In this paper, we propose the Model Instruction Weakness Value (MIWV) as a novel metric to quantify the importance of instruction data in enhancing model's capabilities. The MIWV metric is derived from the discrepancies in the model's responses when using In-Context Learning (ICL), helping identify the most beneficial data for enhancing instruction tuning performance. Our experimental results demonstrate that selecting only the top 1% of data based on MIWV can outperform training on the full dataset. Furthermore, this approach extends beyond existing research that focuses on data quality scoring for data selection, offering strong empirical evidence supporting the effectiveness of our proposed method.

1 Introduction

With the rapid development of natural language processing, LLMs have been able to perform a wide variety of complex language tasks, including writing, translation, conversation, etc (Chiang et al. 2023; Schaeffer, Miranda, and Koyejo 2024; Lin et al. 2024; Liu et al. 2023; Cheng et al. 2024). Representative models, such as OpenAI's O1 and GPT-4, have the advanced capabilities to rapidly learn from vast amounts of data and produce responses that closely align with task requirements. However, with the diversification of application scenarios, solely relying on large-scale data pretraining is no longer sufficient to meet all specific needs. One effective and economical solution is instruction tuning (Xu et al. 2023; Yu et al. 2024; Shu et al. 2023; Tang et al. 2024), which significantly enhances the model's performance on specific tasks by adjusting the model's response to task instructions. This approach allows the LLM to better understand user intent, thereby generating more context-relevant and personalized responses to align with instructions.

Most of the work (Wei et al. 2024; Taori et al. 2023; Wang et al. 2023; Mukherjee et al. 2023) related to instruction tuning has focused on collecting larger, more diverse, and more complex datasets, which costs a great quantity of manpower and material resource, but with low effectiveness (Aghajanyan et al. 2021; Tang et al. 2022). Moreover, blindly increasing dataset size does not ensure desired results, as it can introduce noise and redundancy, hindering the improvement of instruction tuning capabilities (Zhang et al. 2023). Therefore, the key is to adopt an effective data selection strategy when performing instruction tuning, which focus on the quality of data rather than the sheer quantity (Ye et al. 2025; Li et al. 2024b). Recently, the success of (Liu et al. 2024a; Xia et al. 2024; Xie et al. 2023) have profoundly validated this by selecting samples that enhance the model's reasoning capabilities.

In this paper, we propose a universal data selection strategy, which uses in-context learning (ICL) to find high-quality instruction data from the source dataset that is conducive to enhancing the performance of the model. Given that different pretrained LLMs excel in various task domains and have differing capability scopes, our proposed method is applicable to each model and can select the data that best enhances instruction tuning performance for the given LLM. For each sample in the instruction dataset, we first retrieve the most relevant other sample as its one-shot example. We then assess the LLM's ICL performance on prompts presented with and without one-shot examples. If the LLM's performance with one-shot examples is worse than without, it indicates that the LLM lacks the fundamental abilities to respond to this type of instruction effectively. This type of instruction sample acts as high-quality data that is beneficial to enhancing the model's performance.

Furthermore, we propose the Model Instruction Weakness Value (MIWV) metric, which evaluates the effect of instruction samples on enhancing the model's capabilities by calculating the loss difference between the LLM responses to the prompt with and without the one-shot example. The higher the MIWV, the more LLM's capabilities are improved by

*Corresponding Author.

learning the instruction sample. It is noteworthy that we consider the situation where there is no instruction in the source dataset strikingly similar to the given sample, because an irrelevant one-shot example may have negative impact on the response to that instruction. In this case, a high MIWV indicates that the sample is more likely to be chosen, thereby ensuring the diversity of our high-quality data subset. Consequently, a varied instruction-tuning subset can be derived from the source instruction dataset by prioritizing the important samples according to the metrics, which enhances the performance of the LLM.

We conduct instruction tuning experiments on the Alpaca (Taori et al. 2023) and WizardLM (Xu et al. 2023) datasets. Our model outperforms the model trained on the full-scale dataset while utilizing only 1% of the data. Furthermore, we evaluate our method on widely recognized benchmarks, including the Open LLM Leaderboard (Han et al. 2025) and Alpaca Eval (Dubois et al. 2024; Li et al. 2023a). The experimental results demonstrate that our proposed method enhances the performance of the model for instruction tuning. Our contributions are as follows:

1. We propose a universal data selection method based on the computation of sample importance, which can be applied to all LLMs. It is simple, efficient, and fully automated, with no need for model training and dependence on external LLMs.
2. We propose a novel metric named Model Instruction Weakness Value (MIWV) to quantify the importance of instruction samples in enhancing LLM's capabilities. This metric provides a guide to selecting the most beneficial samples from a vast pool of instruction data for improving model's performance. To the best of our knowledge, our work is the first to design a quantitative evaluation metric by leveraging the inherent contextual learning capabilities within instruction data.
3. The experimental results show that instruct-tuning the LLM with a small but high-quality dataset from our data selection strategy significantly enhances its capabilities, which demonstrates the effectiveness and superiority of our method. Moreover, our method achieves better outcomes compared to other advanced approaches.

2 Related Works

2.1 Instruction Tuning

Instruction tuning as a widely utilized training method, allows the model to effectively follow instructions within a specific domain (Liu et al. 2024b; Longpre et al. 2023; Wang et al. 2024a; Qin et al. 2024), ensuring that its responses align with the expected knowledge within that domain. Recent research has focused on developing technical innovations to improve the performance of LLMs, with a specific focus on their abilities to generalize to unfamiliar instructions. Models such as OpenAI's GPT-4 (Achiam et al. 2023) and GPT-4o (Hurst et al. 2024) are trained through instruction tuning and reinforcement learning to produce responses that are more suitable for specific scenarios and favored by users. Since OpenAI has not yet released the source code of advanced models, most of the current work such as Alpaca (Taori et al. 2023) and WizardLM (Li et al. 2024b; Kung et al. 2023; Touvron et al. 2023) adopt LLM from the LLaMA series for instruction tuning. The Qwen2.5 series models (Yang et al. 2024) perform instruction tuning on larger datasets, laying a solid foundation for the application of reasoning capabilities.

2.2 Instruction Data Selection

Although instruction tuning is effective, it also has many challenges: (1) high-quality instruction datasets are required, and existing instruction datasets are usually limited in quantity, diversity, and creativity; (2) continuously increasing the number of tasks and the amount of data may enhance the performance of instruction tuning, but it also consumes a lot of time and resources. InstructMining (Cao et al. 2023) introduces a set of carefully selected metrics for evaluating the quality of instruction samples and applies a statistical regression model to select the high-quality subset, but does not provide performance comparable to models trained using the full data. INSTAG (Lu et al. 2023) and Alpagasus (Chen et al. 2024) utilize ChatGPT to tag the instruction data to ensure diversity and complexity. QDIT (Bukharin and Zhao 2023) and Deita (Liu et al. 2024d) offer the methods to simultaneously control dataset diversity and quality, enabling the study of their effects on instruction tuning. However, these approaches require the use of an additional model to filter high-quality data. RECOST (Zhang et al. 2024a) selects data by calculating differences in conditional entropy, but it relies on external knowledge data. Additionally, some research filters high-quality data using the fine-tuning model itself. For example, SelectIT (Liu et al. 2024c) evaluates sample quality via multiple inferences, deriving token probability distributions to compute a quality score. DiverseEvol (Wu et al. 2023) introduces a self-evolving mechanism that enables the model to proactively sample more effective subsets and iteratively enhance the training set to improve performance.

3 Methodology

In this section, we propose a universal data selection method based on sample importance, which is calculated by retrieving one-shot example ICL. The framework of our method, as demonstrated in Figure 1, consists of three key steps: one-shot example retrieval, computation of sample importance, and high-quality data selection. The details of each step are introduced as follows.

3.1 One-Shot Example Retrieval

In this step, we first compute vector embeddings for all samples in the instruction dataset. Then, for each sample, we find the most similar one among the others (excluding itself). Specifically, for an initial instruction dataset D, we define $x = map(Instruction, [Input])$ as the complete instruction input, y as the corresponding response, and the map function is aligned with the original target dataset. Thus, D consists of n instruction-response pairs $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$. With embedding model

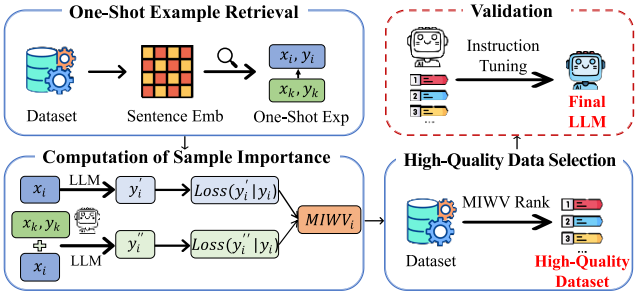


Figure 1: Overview of our proposed method.

$E(\cdot)$, each token of sample x_i is represented as:

$$\mathbf{h}_i^1, \mathbf{h}_i^2, ..., \mathbf{h}_i^Q = E(x_i^1, x_i^2, ..., x_i^Q), \tag{1}$$

where Q is the number of tokens in the sample. Then, we get the representation $\mathbf{h}_i$ as follows:

$$\mathbf{h}_i = \frac{\sum_{q=1}^{Q} \mathbf{h}_i^q}{Q}. \tag{2}$$

For the x_i and another sample x_j (where $j \neq i$), the cosine similarity $sim(\mathbf{h}_i, \mathbf{h}_j)$ between the two can be calculated by their respective corresponding vectors $\mathbf{h}_i$ and $\mathbf{h}_j$. The specific calculation formula is as follows:

$$sim(\mathbf{h}_i, \mathbf{h}_j) = \frac{\mathbf{h}_i \cdot \mathbf{h}_j}{\|\mathbf{h}_i\| \|\mathbf{h}_j\|}. \tag{3}$$

Thus, we calculate the similarity between x_i and all other samples based on Eq. 3, and find the sample x_k that is most similar to x_i by the maximum similarity value:

$$k = arg \max_{j \neq i} \left(sim\left(\mathbf{h}_i, \mathbf{h}_j\right) \right), j \in \{1, 2, ..., n\}. \tag{4}$$

Finally, we define (x_k, y_k) as the one-shot example of (x_i, y_i).

3.2 Computation of Sample Importance

It is challenging to assess the quality of an instruction sample without the assistance of human experts, let alone applying it to all LLMs. Therefore, a quantitative metric to measure the importance of a sample based on the LLMs is essential. We boldly introduce the MIWV metric to quantatively evaluate the importance of each instruction sample in enhancing model's performance. The MIWV of each sample is defined as the loss difference between the LLM responses to the prompt with and without one-shot examples. During instruction tuning, the loss of the instruction-response pair is calculated by continuously predicting the next token based on the previous words when the instruction is given.

$$L_\theta\left(y_i|x_i\right) = -\frac{1}{A} \sum_{a=1}^{A} \log p\left(y_i^a | x_i, y_i^1, ..., y_i^{a-1}\right), \tag{5}$$

where A represents the length of the groundtruth answer y_i, and y_i^a denotes its a-th token. $L_\theta\left(y_i|x_i\right)$ is used to evaluate the degree of challenge encountered by the LLM in generating an answer to a given instruction. However, it does not directly reflect the actual response abilities of LLM to the instruction sample, since the performance of LLM may be affected by the inherent characteristics of the instruction, and its potential capabilities may not be fully stimulated. More accurately, in order to measure the model's capabilities to understand and respond to the instruction, we introduce one-shot examples into the prompt to enrich the input information, which can stimulate LLM's capabilities, as follows:

$$C = \text{Prompt}\left(x_k, y_k\right), \tag{6}$$

$$L_\theta(y_i|x_i, C) = -\frac{1}{A} \sum_{a=1}^{A} \log p\Big(y_i^a \mid x_i, C, y_i^1, \ldots, y_i^{a-1}\Big), \tag{7}$$

where C is the one-shot prompt of the instruction sample (x_i, y_i)). We refer to $L_\theta\left(y_i|x_i, C\right)$ as the prompt loss. It measures the capabilities of the model to make the correct response when prompted by the similar sample. Consequently, in order to reflect the model's weakness in dealing with the instruction sample, MIWV is calculated as

$$\text{MIWV}\left(x_i, y_i\right) = L_\theta\left(y_i|x_i, C\right) - L_\theta\left(y_i|x_i\right). \tag{8}$$

A high MIWV value indicates that samples elicit weak responses from the LLM, making them valuable for LLM's enhancement of capabilities.

3.3 High-Quality Data Selection

In order to obtain the final LLM that performs better on specific tasks, it's essential to acquire a high-quality dataset for model instruction tuning. The high-quality dataset is made up of important instruction samples with high MIWV value.

We have used MIWV to evaluate the importance of each sample. In this section, we rank all instruction samples based on the calculated MIWV values and prioritize those with higher MIWV values. Consequently, the samples with the highest MIWV values form a high-quality subset of data. This subset is then used for instruction tuning to validate the model's abilities improvement.

4 Experiments

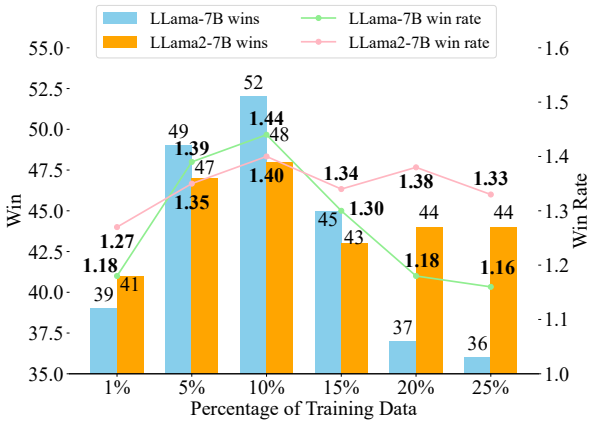
4.1 Datasets

Train Datasets. To verify the effectiveness of our method, we conduct experiments on two datasets: Alpaca (Taori et al. 2023), and WizardLM (Xu et al. 2023), which contain 52002 and 63655 instruction samples, respectively. The dataset details are in Appendix. A.

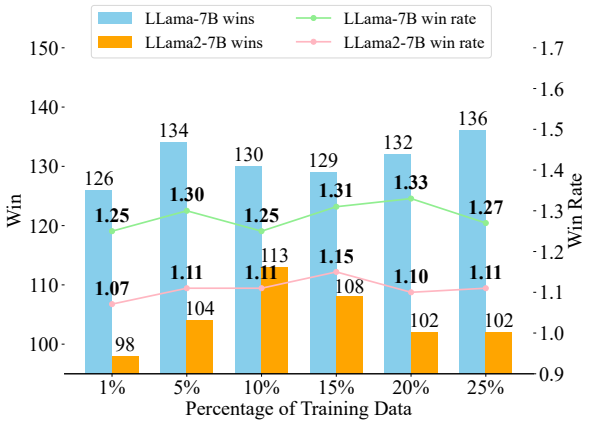
Test Datasets. We use five distinct test datasets based on existing literature: Vicuna (Chiang et al. 2023), Koala (Vu et al. 2023), WizardLM (Xu et al. 2023), Self-instruct (Wang et al. 2022a), and LIMA (Zhou et al. 2024). These datasets total 1,030 carefully designed instruction samples.

4.2 Experimental Setting

We implement MIWV using PyTorch 2.0.1 on a Linux server equipped with 984GB RAM, Intel Xeon Platinum



(a) The distribution of the number of win and win rates for the Alpaca instruction dataset on different models under the Vicuna test set.



(b) The distribution of the number of win and win rates for the WizardLM instruction dataset on different models under the Sinstruct test set.

Figure 2: Comparing our models trained on selected data with full data. Both (a) and (b) use GPT-4 as the judge.

8369B CPU @ 2.90GHz, and Nvidia A100 Tensor Core 80GB GPUs. Bge-en-large (Xiao et al. 2024) serves as the embedding model for one-shot sample retrieval. We employ LLaMA-7B and LLaMA2-7B/13B following the training parameters in (Taori et al. 2023), using the Alpaca codebase. All experiments are repeated three times with arithmetic mean results reported. The detailed training configurations are provided in Appendix. A.

4.3 Evaluation Metrics

Considering the disadvantages of manual evaluation, we employ three metrics (Chang et al. 2024; Chen et al. 2024; Zhou et al. 2024) to assess the effectiveness of our method, including (1) Pair-wise Comparison, (2) Open LLM Leaderboard, and (3) Alpaca Eval. The details of the evaluation metrics are listed in Appendix. B.

4.4 Main Results

We apply the data selection method on two train datasets and five test datasets, and use the selected subsets to fine-tune the baseline LLMs, including LLaMA-7B and LLaMA2-7B pretrained models. As shown in Figure 2(a), it can be observed that the Alpaca 1% model (trained on the top 1% of data selected by MIWV) significantly outperforms the officially trained Alpaca model (trained on the complete dataset) on the Vicuna test set. By comparing the models obtained from different data ratios, the model trained with 10% data achieves the highest win rate. Figure 2(b) further illustrates that on the Sinstruct test set, the WizardLM 1% model, which is trained on the top 1% of data selected by MIWV, also outperforms the WizardLM model trained on the full dataset. In experiments using the LLaMA-7B pretrained model, the WizardLM 20% model has the best win rate, while in experiments with the LLaMA2-7B pretrained model, the WizardLM 15% model exhibits the best performance. It is noteworthy that as the proportion of data used for training increases, the overall win rate exhibits a declining trend. It suggests that the data may interfere with each other or contain harmful noise, further validating the effectiveness of our MIWV method for data selection.

Table 1 presents the results of three automatic evaluation methods. It utilizes the PairWise Win Rate to directly compare our models with the corresponding models trained on the full dataset. These values greater than 1.0 for this metric indicates that our models outperform full-dataset models. Additionally, the performance of our models and the baseline models on the Huggingface Open LLM Leaderboard and the AlpacaEval Leaderboard are also illustrated in the table. It reveals that models trained on 1%, 5%, 10%, and 15% of the data surpass the models trained on the full dataset in benchmark tests across both the LLaMA2-7B and LLaMA2-13B models. These results validate the effectiveness of our method.

Furthermore, the experimental results of the MIWV method on the classic multi-task natural language processing dataset NIV2 (Wang et al. 2022b) further validate its effectiveness. For details, we list on Appendix. D.

4.5 Comparison with Other Methods

We compare the performance and efficiency of our method with five other widely accepted studies using LLaMA2-7B on the Alpaca dataset, including IFD Score (Li et al. 2024b), SelectIT (Liu et al. 2024c), Superfiltering (Li et al. 2024a), Alpagasus (Chen et al. 2024), Deita (Liu et al. 2024d), DiverseEvol (Wu et al. 2023), Nuggets (Li et al. 2023b) and RECOST (Zhang et al. 2024a).

As shown in Table 2, the win rate of our method on the test set is significantly ahead of the latest works and is second only to Superfiltering in terms of efficiency. For the IFD Score, both the SelectIT and DiverseEvol approaches require model training, resulting in lower efficiency and poorer performance compared to our method. Nuggets and RECOST rely on a specific set of tasks and external knowledge, respectively, which introduce scoring biases. This is

Dataset/ Base Model	MIWV Ratio (Size)	Pairwise Win Rate↑	Huggingface Open LLM Leaderboard↑ Average	ARC	HellaSwag	MMLU	TruthfulQA	Alpaca Eval↑
Alpaca/ LLaMA2-7B	100%	1.000	55.25	54.35	78.65	47.02	40.98	27.75
	1% (520)	1.127	56.17	57.25	78.28	47.84	41.32	**39.50**
	5% (2600)	1.214	56.91	58.82	79.31	48.83	40.98	**39.87**
	10% (5200)	1.228	**57.36**	58.53	79.90	49.26	41.74	-
	15% (7800)	**1.248**	57.08	58.93	78.73	48.40	42.27	-
Alpaca/ LLaMA2-13B	100%	1.000	58.78	57.59	81.98	54.05	41.49	35.00
	1% (520)	1.063	60.36	62.80	82.03	54.97	41.64	**41.30**
	5% (2600)	1.160	**61.48**	63.46	83.82	55.82	42.82	**48.24**
	10% (5200)	1.200	60.74	62.69	82.73	55.06	42.47	-
	15% (7800)	**1.256**	61.11	62.82	82.76	55.56	43.28	-
WizardLM/ LLaMA2-7B	100%	1.000	55.02	58.61	73.32	41.21	46.92	59.25
	1% (636)	1.048	55.45	60.14	75.54	40.48	45.37	**60.12**
	5% (3182)	1.096	55.58	61.49	75.32	40.76	45.98	**61.37**
	10% (6365)	1.114	55.98	60.47	76.47	40.25	46.72	-
	15% (9548)	**1.153**	**57.07**	60.85	76.98	41.70	48.73	-
WizardLM/ LLaMA2-13B	100%	1.000	60.23	61.32	80.39	51.00	48.19	67.95
	1% (636)	1.043	60.30	61.73	80.25	50.67	48.54	**68.32**
	5% (3182)	1.050	60.41	61.92	80.78	50.25	48.67	**69.81**
	10% (6365)	1.107	60.49	61.56	80.46	50.80	49.12	-
	15% (9548)	**1.118**	**61.00**	62.34	80.84	51.23	49.58	-

Table 1: Comparison of our method with four data selection ratios (1%, 5%, 10%, 15%, 100%) when fine-tuning two LLMs (LLaMA2-7B/13B) on Alpaca and WizardLM datasets. The finetuned models are evaluated by the Pair-wise win rate (compared to the baseline model finetuned on 100% data), Open LLM Leaderboard, and AlpacaEval.

the main reason their win rates are lower than those of our method. The performance of Alpagasus and Deita are not only inferior to our method, but their reasoning efficiency is also constrained by ChatGPT API rate limits. As for Superfiltering, although the method is more efficient, inconsistent models used in data selection and instruction fine-tuning can have a negative impact on the win rate.

4.6 Ablation Study

We conduct extensive ablation experiments from two aspects to validate the effectiveness of our method: different data selection strategies and embedding models for instruction sample retrieval.

Ablation on Data Selection strategy. In addition to our method, we systematically investigate three other data selection strategies: Data Randomly Selected, Data Selected with High Prompt Loss, and Data Selected with Low MIWV. For each data selection strategy, we set four different data ratios: 1%, 5%, 10%, and 15%. We train the LLaMA-7B models on the Alpaca dataset for each of these ratios and compare their performance against the official Alpaca model. The results are shown in Figure 3.

Data Randomly Selected. We train the models using randomly chosen data. As shown in Figure 3(a), the models trained with random data always perform worse than the official Alpaca model.

Data Selected with High Prompt Loss. We select data for instruction tuning based on high Prompt Loss. As shown in Figure 3(a), the model trained with these data performs worse than the official Alpaca model. This indicates that the high Prompt Loss fails to select truly valuable data.

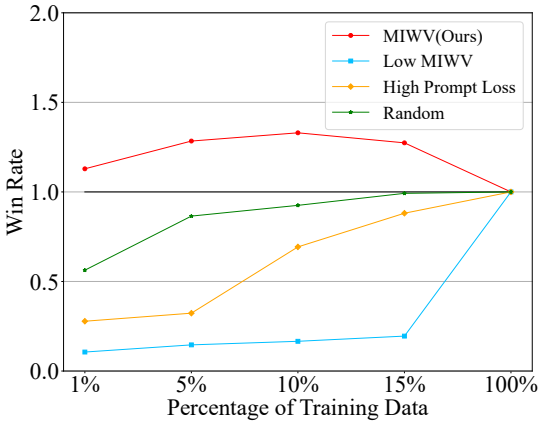
Metric	Win Rate ↑				Time
Selection Scale	**1%**	**5%**	**10%**	**15%**	min
IFD Score	0.794	0.853	0.761	0.927	161
SelectIT	0.954	1.073	<u>1.151</u>	1.183	184
Superfiltering	0.972	1.133	1.101	<u>1.193</u>	**8**
Alpagasus	0.982	1.028	1.119	1.170	120
Deita	1.009	1.092	1.032	1.013	282
DiverseEvol	1.018	<u>1.142</u>	1.137	1.165	300
Nuggets	1.037	1.041	1.124	1.050	210
RECOST	<u>1.092</u>	1.138	1.147	1.110	152
MIWV (Ours)	**1.119**	**1.211**	**1.178**	**1.234**	<u>85</u>

Table 2: Comparison win rates of other methods. All comparisons are performed by GPT-4 on the WizardLM test set. The time shown represents the time used for data selection.

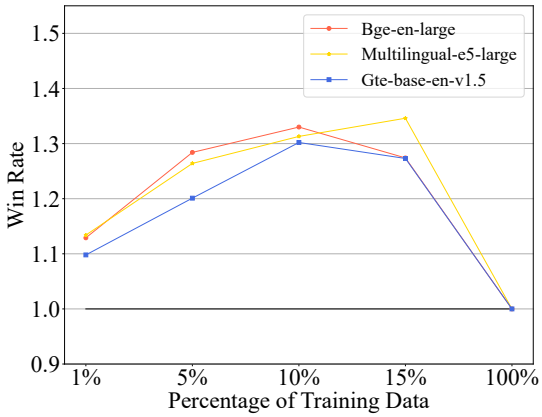
Data Selected with Low MIWV. We select data with low MIWV for instruction tuning. As shown in Figure 3(a), the model trained with low MIWV data has the lowest win rate, which is also worse than the official Alpaca model.

The results above show that the other three data selection strategies all fail to select valuable data to enhance the model's performance, while our method achieves the goal.

Ablation on Embedding Model. In order to ensure the reliability and generalization of the experiments, we extract one-shot examples across various embedding models, including Bge-en-large, Multilingual-e5-large (Wang et al. 2024b), and Gte-base-en-v1.5 (Zhang et al. 2024b). These models, with different scales and multilingual capabilities,



(a) The performance of various data selection strategies.



(b) The performance of using various embedding models.

Figure 3: The comparison of the win rate between models and the official Alpaca model of ablation study.

increase the experiment's comprehensiveness. We create high-quality instruction subsets with 1%, 5%, 10%, and 15% ratios from the Alpaca dataset for instruction tuning on LLaMA-7B, as shown in Figure 3(b). Our findings indicate that our method effectively enhances model's performance regardless of the embedding model. Specifically, models with 5% and 10% subsets using Bge-en-large and the 15% subset with Multilingual-e5-large performed best. Although the subset chosen with Gte-base-en-v1.5 shows a slightly lower win rate, it still significantly outperforms the model trained on the full dataset.

4.7 Analysis

Cross-Series Models Effectiveness. To further validate the effectiveness and versatility of our method, we apply it to Qwen2.5-7B and Qwen2.5-14B models, which has distinct architectures and capabilities compared to LLaMA/LLaMA2. Figure 4 shows the distribution of win numbers and win rates for the Alpaca instruction dataset under the LIMA test set evaluated by GPT-4. We can find that our method demonstrates exceptional performance on the Qwen2.5 series models. Specifically, models trained with a small amount of high-quality filtered data consistently outperform those trained on the original complete dataset, underscoring the broad applicability of our approach.

ICL-Guided Effectiveness. To further demonstrate the role of ICL in the data selection process, we design a comparative experiment based on the "IFD Score". Specifically, we replace the clustering step in the original "IFD Score" method with an ICL approach that extracts embeddings from instruction samples, retrieves one-shot examples for each sample, and computes IFD scores via model inference. We select the 1% subset of the Alpaca instruction dataset and conduct experiments on the LLaMA-7B. The comparison results of the two methods are demonstrated in Table 3. The experimental results show that the model trained with data selected by ICL performs better than the original "IFD Score",

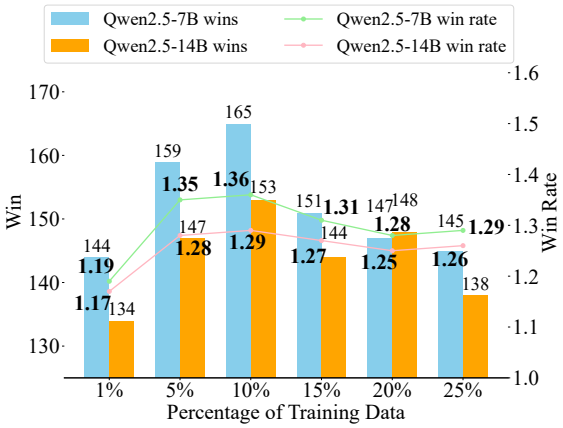


Figure 4: Comparison of Qwen2.5 models trained on selected and full data.

with an overall win rate of 1.017 significantly higher than 0.939. Furthermore, the proposed MIWV approach achieves the highest win rate of 1.140, significantly surpassing both methods. These results emphasize the effectiveness and superiority of context learning-assisted data selection and the MIWV metric.

4.8 Case Study

Figure 5 presents a case study focused on mathematical abilities, utilizing the LLaMA2-7B model trained with 1% of the Alpaca dataset (Alpaca-1%) and with the full Alpaca dataset (Alpaca-100%). The results demonstrate that the Alpaca-1% model successfully provides a correct answer, whereas the Alpaca-100% model fails to do so.

4.9 Data Characteristics

Distribution Characteristics. We visualize the 2D distribution of high-dimensional embeddings from the Alpaca

Test Datasets	Vicuna			Koala			WizardLM			Sinstruct			LIMA		
GPT4 Eval	win↑	tie↑	lose↓	win↑	tie↑	lose↓	win↑	tie↑	lose↓	win↑	tie↑	lose↓	win↑	tie↑	lose↓
IFD Score	30	12	38	56	58	66	62	49	107	86	67	99	122	62	116
ICL + IFD Score	34	15	31	59	**61**	60	74	55	89	92	73	87	129	68	103
MIWV (Ours)	**39**	**16**	**25**	**76**	52	**52**	**97**	**62**	**59**	**100**	**81**	**71**	**136**	**74**	**90**

Table 3: Comparison of the performance of "IFD Score" with introducing In-Context Learning strategy and the original "IFD Score" on five test sets, using GPT-4 as the judge.

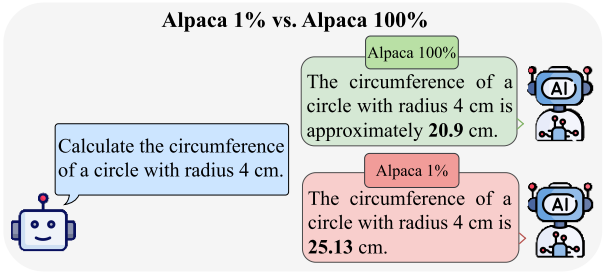


Figure 5: Case Study on Mathematical Capabilities: LLaMA2-7B Models of Alpaca 1% and Alpaca 100%.

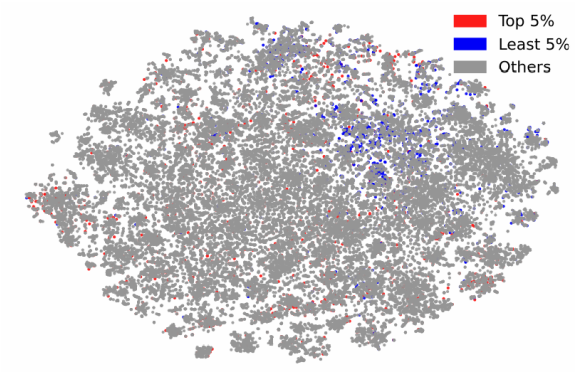


Figure 6: Visualization using t-SNE on instruction embeddings from the Alpaca dataset.

dataset using t-SNE. In Figure 6, red, blue, and gray points denote the top 5%, least 5%, and remaining other samples ranked by MIWV, respectively. The visualization reveals that higher MIWV samples exhibit uniform distribution across the instruction spectrum while lower MIWV samples cluster in specific regions. This pattern indicates that selected data should evenly cover the instruction set distribution to maximize diversity. The clustering of samples in the embedding space indicates that MIWV remains a meaningful discriminative indicator even in the reduced-dimensional representation. Notably, Lower MIWV samples typically involve basic tasks like editing punctuation, words, or simple sentences. In contrast, higher MIWV samples include both basic and more complex tasks, such as storytelling and explaining phenomena. This diversity is critical for LLM performance, enabling deeper understanding, adaptability, and handling of complex language structures.

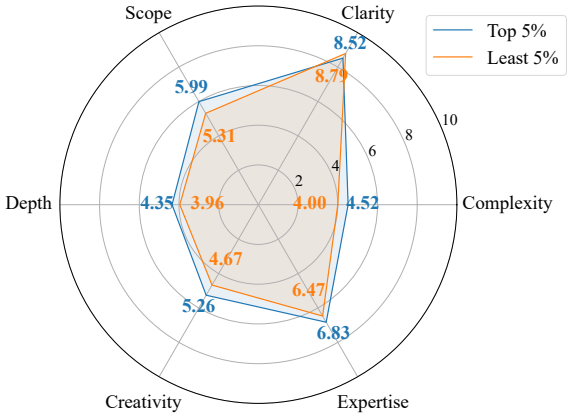


Figure 7: Quality scores for the top and least 5% of data instances in the Alpaca dataset evaluated by GPT-4.

Quality Characteristics. To validate MIWV's alignment with data quality, we use GPT-4 to evaluate 100 randomly selected top and the least 5% samples across six dimensions. As shown in Figure 7, higher MIWV samples outperform in Complexity, Scope, Depth, Creativity, and Expertise while maintaining comparable Clarity. This multi-dimensional analysis confirms MIWV's effectiveness in balancing instructional diversity and cognitive challenges, critical for robust LLM training. Integrating spatial distribution and quality metrics highlight MIWV as a robust criterion. Appendix. E shows the details of six dimensions.

5 Conclusion

In this paper, we propose a universal high-quality data selection method based on a novel metric called Model Instruction Weakness Value (MIWV), which quantifies instruction sample importance through response discrepancies in In-Context Learning (ICL). Extensive experimental results consistently demonstrate that the LLM instruction-tuned on a small amount of high-quality subset achieves results comparable to or even surpassing those tuned on the full-scale dataset, which proves the effectiveness and superiority of our method. This finding also highlights the potential of our method for the best use of limited resources, offering a new approach for economical and efficient data utilization in model training. Furthermore, the application of ICL for data selection is a relatively novel concept that is expected to inspire future research in data selection methodologies.

Acknowledgments

This work was supported by JST CREST Grant Number JPMJCR21M2, including the AIP Challenge Program.

References

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

Aghajanyan, A.; Gupta, A.; Shrivastava, A.; Chen, X.; Zettlemoyer, L.; and Gupta, S. 2021. Muppet: Massive Multi-task Representations with Pre-Finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 5799–5811.

Bukharin, A.; and Zhao, T. 2023. Data diversity matters for robust instruction tuning. *arXiv preprint arXiv:2311.14736*.

Cao, Y.; Kang, Y.; Wang, C.; and Sun, L. 2023. Instruction Mining: When Data Mining Meets Large Language Model Finetuning. *arXiv preprint arXiv:2307.06290*.

Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Yang, L.; Zhu, K.; Chen, H.; Yi, X.; Wang, C.; Wang, Y.; et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3): 1–45.

Chen, L.; Li, S.; Yan, J.; Wang, H.; Gunaratna, K.; Yadav, V.; Tang, Z.; Srinivasan, V.; Zhou, T.; Huang, H.; and Jin, H. 2024. AlpaGasus: Training a Better Alpaca with Fewer Data. In *The Twelfth International Conference on Learning Representations, ICLR*.

Cheng, X.; Zhu, Z.; Li, H.; Li, Y.; Zhuang, X.; and Zou, Y. 2024. Towards multi-intent spoken language understanding via hierarchical attention and optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 16, 17844–17852.

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

Dubois, Y.; Li, C. X.; Taori, R.; Zhang, T.; Gulrajani, I.; Ba, J.; Guestrin, C.; Liang, P. S.; and Hashimoto, T. B. 2024. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36.

Han, J.; Liu, H.; Fang, J.; Tan, N.; and Xiong, H. 2025. Automatic Instruction Data Selection for Large Language Models via Uncertainty-Aware Influence Maximization. In *THE WEB CONFERENCE 2025*.

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

Kung, P.; Yin, F.; Wu, D.; Chang, K.; and Peng, N. 2023. Active Instruction Tuning: Improving Cross-Task Generalization by Training on Prompt Sensitive Tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP*, 1813–1829.

Li, M.; Zhang, Y.; He, S.; Li, Z.; Zhao, H.; Wang, J.; Cheng, N.; and Zhou, T. 2024a. 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), ACL*, 14255–14273.

Li, M.; Zhang, Y.; Li, Z.; Chen, J.; Chen, L.; Cheng, N.; Wang, J.; Zhou, T.; and Xiao, J. 2024b. 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)*, 7595–7628.

Li, X.; Zhang, T.; Dubois, Y.; Taori, R.; Gulrajani, I.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023a. Alpacaeval: An automatic evaluator of instruction-following models.

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

Lin, B. Y.; Deng, Y.; Chandu, K.; Brahman, F.; Ravichander, A.; Pyatkin, V.; Dziri, N.; Bras, R. L.; and Choi, Y. 2024. WILDBENCH: Benchmarking LLMs with Challenging Tasks from Real Users in the Wild. *arXiv preprint arXiv:2406.04770*.

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Liu, F.; Wang, X.; Yao, W.; Chen, J.; Song, K.; Cho, S.; Yacoob, Y.; and Yu, D. 2024b. MMC: Advancing Multimodal Chart Understanding with Large-scale 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), NAACL*, 1287–1310.

Liu, L.; Liu, X.; Wong, D. F.; Li, D.; Wang, Z.; Hu, B.; and Zhang, M. 2024c. Selectit: Selective instruction tuning for large language models via uncertainty-aware self-reflection. *arXiv preprint arXiv:2402.16705*.

Liu, P.; Liu, Z.; Gao, Z.-F.; Gao, D.; Zhao, W. X.; Li, Y.; Ding, B.; and Wen, J.-R. 2023. Do emergent abilities exist in quantized large language models: An empirical study. *arXiv preprint arXiv:2307.08072*.

Liu, W.; Zeng, W.; He, K.; Jiang, Y.; and He, J. 2024d. What Makes Good Data for Alignment? A Comprehensive Study of Automatic Data Selection in Instruction Tuning. In *The Twelfth International Conference on Learning Representations, ICLR*.

Longpre, S.; Hou, L.; Vu, T.; Webson, A.; Chung, H. W.; Tay, Y.; Zhou, D.; Le, Q. V.; Zoph, B.; Wei, J.; et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, 22631–22648. PMLR.

Lu, K.; Yuan, H.; Yuan, Z.; Lin, R.; Lin, J.; Tan, C.; Zhou, C.; and Zhou, J. 2023. # instag: Instruction tagging for analyzing supervised fine-tuning of large language models. In

The Twelfth International Conference on Learning Representations.

Mukherjee, S.; Mitra, A.; Jawahar, G.; Agarwal, S.; Palangi, H.; and Awadallah, A. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

Qin, Y.; Yang, Y.; Guo, P.; Li, G.; Shao, H.; Shi, Y.; Xu, Z.; Gu, Y.; Li, K.; and Sun, X. 2024. Unleashing the power of data tsunami: A comprehensive survey on data assessment and selection for instruction tuning of language models. *arXiv preprint arXiv:2408.02085*.

Schaeffer, R.; Miranda, B.; and Koyejo, S. 2024. Are emergent abilities of large language models a mirage? *Advances in Neural Information Processing Systems*, 36.

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

Tang, P.; Yang, C.; Xing, T.; Xu, X.; Jiang, R.; and Sezaki, K. 2024. Instruction-Tuning Llama-3-8B Excels in City-Scale Mobility Prediction. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Human Mobility Prediction Challenge*, 1–4.

Tang, T.; Li, J.; Zhao, W. X.; and Wen, J.-R. 2022. Mvp: Multi-task supervised pre-training for natural language generation. *arXiv preprint arXiv:2206.12131*.

Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; and Alpaca, T. H. S. 2023. An Instruction-Following LLaMA Model.

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

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

Wang, J.; Zhang, B.; Du, Q.; Zhang, J.; and Chu, D. 2024a. A Survey on Data Selection for LLM Instruction Tuning. *arXiv preprint arXiv:2402.05123*.

Wang, L.; Yang, N.; Huang, X.; Yang, L.; Majumder, R.; and Wei, F. 2024b. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

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

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2023. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Wang, Y.; Mishra, S.; Alipoormolabashi, P.; Kordi, Y.; Mirzaei, A.; Naik, A.; Ashok, A.; Dhanasekaran, A. S.; Arunkumar, A.; Stap, D.; et al. 2022b. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. In *EMNLP*.

Wei, Y.; Wang, Z.; Liu, J.; Ding, Y.; and Zhang, L. 2024. Magicoder: Empowering Code Generation with OSS-Instruct. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 52632–52657. PMLR.

Wu, S.; Lu, K.; Xu, B.; Lin, J.; Su, Q.; and Zhou, C. 2023. Self-evolved diverse data sampling for efficient instruction tuning. *arXiv preprint arXiv:2311.08182*.

Xia, M.; Malladi, S.; Gururangan, S.; Arora, S.; and Chen, D. 2024. LESS: Selecting Influential Data for Targeted Instruction Tuning. In *International Conference on Machine Learning (ICML)*.

Xiao, S.; Liu, Z.; Zhang, P.; Muennighoff, N.; Lian, D.; and Nie, J. 2024. C-Pack: Packed Resources For General Chinese Embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, 641–649.

Xie, S. M.; Santurkar, S.; Ma, T.; and Liang, P. S. 2023. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36: 34201–34227.

Xu, C.; Sun, Q.; Zheng, K.; Geng, X.; Zhao, P.; Feng, J.; Tao, C.; and Jiang, D. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Ye, Y.; Huang, Z.; Xiao, Y.; Chern, E.; Xia, S.; and Liu, P. 2025. LIMO: Less is More for Reasoning. arXiv:2502.03387.

Yu, Y.; Zhuang, Y.; Zhang, J.; Meng, Y.; Ratner, A. J.; Krishna, R.; Shen, J.; and Zhang, C. 2024. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in Neural Information Processing Systems*, 36.

Zhang, Q.; Zhang, Y.; Wang, H.; and Zhao, J. 2024a. Recost: External knowledge guided data-efficient instruction tuning. *arXiv preprint arXiv:2402.17355*.

Zhang, X.; Zhang, Y.; Long, D.; Xie, W.; Dai, Z.; Tang, J.; Lin, H.; Yang, B.; Xie, P.; Huang, F.; et al. 2024b. mGTE: Generalized Long-Context Text Representation and Reranking Models for Multilingual Text Retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 1393–1412.

Zhang, Y.; Li, Y.; Cui, L.; Cai, D.; Liu, L.; Fu, T.; Huang, X.; Zhao, E.; Zhang, Y.; Chen, Y.; et al. 2023. Siren's song in the AI ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Zhou, C.; Liu, P.; Xu, P.; Iyer, S.; Sun, J.; Mao, Y.; Ma, X.; Efrat, A.; Yu, P.; Yu, L.; et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.