LIMACOST: Data Valuation for Instruction Tuning of Large Language Models

Hyeonseok Moon[1], **Jaehyung Seo**[1], **Seonmin Koo**[1], **Jinsung Kim**[1], **Young-kyoung Ham**[2], **Jiwon moon**[2], **Heuiseok Lim**[1,†]
[1]Department of Computer Science and Engineering, Korea University
[2]Korea Telecom
{glee889,seojae777,fhdahd,jin62304,limhseok}@korea.ac.kr
{youngkyoung.ham,jiwon.moon}@kt.com

Abstract

Instruction tuning (IT) is an effective approach for aligning large language models (LLMs) with human intentions. There is ongoing discourse regarding the data quality for IT. As an effort to find the robust criteria of data quality for IT, we introduce LIMACOST, a data quality measure that exhibits a strong correlation with model performance. LIMACOST utilizes LIMA dataset, which effectiveness in IT has already been validated by several previous works. LIMACOST then estimates the value of a given data by estimating how many LIMA data points might be needed to approximate its gradient. Our experiments reveal that LIMACOST enables effective data selection that derive high alignment performance. We demonstrate that selecting data based on high LIMACOST proves to be more effective than existing data selection strategies.

1 Introduction

Instruction Tuning (IT) is one of the practical strategy for large language models (LLMs) to attain human-interactive capability (Zhang et al., 2023; Longpre et al., 2023; Peng et al., 2023). To achieve more effective alignment, recent studies have highlighted the importance of acquiring high-quality IT data over a large quantity (Chen et al., 2024b; Xia et al., 2024b; Zhou et al., 2023; Wang et al., 2024).

Then, *what data should we choose for IT?* Numerous studies have sought a satisfactory answer to this question (Liu et al., 2024c; Albalak et al., 2024; Chen et al., 2024a), but we have yet to find a definitive answer. These attempts include quantifying data quality using frontier LLMs like ChatGPT (Chen et al., 2024b; Liu et al., 2024c; Bukharin and Zhao, 2023), or establishing explicit quality criteria and manually designed data based on those standards (Zhou et al., 2023; Liu et al., 2024b; Zhao et al., 2024).

We observed that these studies are often conducted in isolation without leveraging existing research resources. Although previous research has provided high-quality datasets and robust performance metrics, current studies on data quality often establish their distinct standards without considering prior works, resulting in a lack of continuity. Notably, several high-quality datasets, such as LIMA (Zhou et al., 2023), and their well-established data quality assets appear underutilized.

In this study, we aim to address the aforementioned question regarding data selection by utilizing previous assets. Specifically, we introduce LIMACOST, a data valuation method that can select high quality data that derive superior alignment performance. The concept behind LIMACOST is straightforward: it identifies data points that significantly contribute to model updates. If a data point yields a complex gradient that is challenging to estimate from existing data, we deduce that it embeds high-quality information.

To achieve this, we utilize the LIMA (Zhou et al., 2023) dataset, known for its credibility and effectiveness in alignment tuning. LIMACOST assesses each data point by determining the number of LIMA data points required to approximate its gradient. If only few LIMA data points are necessary to estimate the gradient of a given data point, the impact from that data point is considered relatively minor. We consider that a high LIMACOST value implies significant effectiveness for alignment tuning.

Our experiments reveal that LIMACOST allows us to estimate the impact of each data on alignment tuning. To validate LIMACOST, we sorted the Alpaca-gpt4 (Peng et al., 2023) and EvolInstruct (Xu et al., 2023) datasets by their LIMACOST scores. We selected the top 1,000 high-score data points and trained LLMs with varying knowledge capacities—namely, Llama-2-7B (Touvron et al., 2023) and Mistral-7B (Jiang et al., 2023)—using

† Corresponding Author

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 12841–12854
November 4-9, 2025

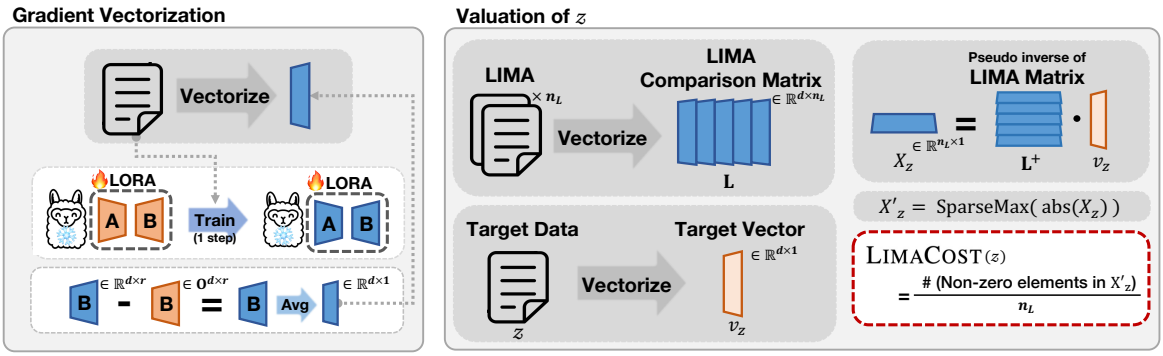

Figure 1: Overall process of LIMACOST

these data. We subsequently evaluated the instruction following performance of the trained models. Our findings confirm that training with high LIMACOST data results in more effective alignment performance than using low LIMACOST data. Additionally, training with high LIMACOST data surpasses traditional strong IT data selection strategies, such as NUGGETS (Li et al., 2023) and SelectIT (Liu et al., 2024a).

These findings demonstrate that LIMACOST serves as a straightforward and effective metric for assessing data quality, offering clear insights into how certain high-quality data points can receive high ratings. We detail the rationale and applicability of our methodology through an extensive analyses.

2 Related Works

IT has garnered significant attention with the advent of LLMs (Longpre et al., 2023; Xu et al., 2023; Liu et al., 2023). Specifically, research over recent years has established a consensus that the quality of instruction tuning is crucial for ensuring alignment (Chen et al., 2024b; Liu et al., 2024c; Wang et al., 2023a; Zhou et al., 2023; Albalak et al., 2024). This research trajectory continues to evolve, with a rich discourse on defining high-quality data and strategies for obtaining it (Gao et al., 2020; Wettig et al., 2024; Lu et al., 2024). Approaches include using model internal knowledge to select high-quality data (Xia et al., 2024b; Li et al., 2024b) and employing GPT evaluations to establish quality benchmarks (Liu et al., 2024c; Chen et al., 2024b; Wettig et al., 2024).

Notably, research presented by LIMA (Zhou et al., 2023) demonstrated that carefully designed 1,000 pieces of data (experts manually devised heuristics indicative of high-quality data (Albalak et al., 2024)) can yield a significantly high instruction-following capability. Moreover, several recent works figured out that selectively training on the small fraction of selected Alpaca data points show more effective than employing the entire 52K dataset (Chen et al., 2024b; Liu et al., 2024c; Wettig et al., 2024; Zhao et al., 2024).

However, we find that these studies have not yet provided compelling explanations. The quality scores evaluated by frontier LLMs do not show a strong correlation with the performance of the trained models (Liu et al., 2024c). Additionally, most other quality scores rely heavily on human-defined heuristics (Albalak et al., 2024; Bommasani et al., 2023; Iyer et al., 2022; Ivison et al., 2023) or remain at the assumption stage that they might benefit model training (Zhao et al., 2024; Wang et al., 2023a). In response, we propose a quality metric that is more directly linked to model performance and allows for objective valuation of each data point.

3 LIMACOST

LIMACOST aims to quantify the influence of each data points on the model during IT process. In other words, LIMACOST serves as a quality measure for each data point z, with higher LIMACOST(z) indicating higher quality. Specifically, we measure the change of the model parameters when trained on each data point (*i.e.* also can be expressed as a gradient), and count the amount of LIMA data needed to estimate that change (*i.e.* gradient). To accomplish this, we define the following processes.

3.1 Gradient Vectorization

To achieve more intuitive quantification, we estimate the gradient obtained from a single data point z by evaluating the change of the model parameters.

We define the initial model state as θ and denote the model state after training on a given data point z as θ_z. We then define the parameter difference between θ_z and θ as ∇_z. To ensure robust estimation, we reload initial model states and optimizer prior to training with each data point. Given that we directly estimate changes in the model parameters, we contend that the choice of optimizer likely does not significantly impact the results.

In training θ with z, we incorporate a LoRA (Hu et al., 2022) structure to achieve two primary benefits. First, it allows for relatively efficient training. By freezing all states except those related to LoRA, we update only at most 1% of the parameters, facilitating an efficient training process. In this work, we set the dimension of LoRA structure r as 8 for more efficient calculation (training only about 0.1% of the whole parameters). Second, as LoRA structure comprises two linear layers (*i.e.* LoRA_A and LoRA_B) and one of which (LoRA_B) is initialized with zero state, we can effectively track the update of model states. We provide more details in the Section 3.4.

Through this process, we obtain the amount of update in the LoRA structure derived by z. Note that $\theta^{loraB} \in \mathbf{0}^{d\times r}$ by initialization and d is the hidden size of θ.

$$\theta_z^{loraB} = \theta^{loraB} + \eta\nabla_z^{loraB} \quad (1)$$

$$= \eta\nabla_z^{loraB} \in \mathbb{R}^{d\times r} \quad (\theta^{loraB} \in \mathbf{0}) \quad (2)$$

As the initial state of LoRA_B is the zero state, we can track the impact of a given data point on the model by only observing the difference of LoRA_B. By calculating the average of these changes, we derive the vector value v_z for z as in Equation (3). We specifically use only the model state of the LoRA structure in the first layer for calculating v_z. We will discuss a case study on this topic in the section.

$$v_z = \text{Avg}(\nabla_z^{loraB}, dim = 1) \in \mathbb{R}^{d\times 1} \quad (3)$$

This approach enables us to vectorize the impact of a data point z on the change of model state. Here, we set the learning rate η to 1e-5. It is important to note that any arbitrary choice of η does not affect the determination of LIMACOST in the subsequent process.

3.2 Lima Comparison Matrix

Using the aforementioned vectorization method, we vectorize all the LIMA (Zhou et al., 2023) data and compute the gradient matrix L for these data. This matrix subsequently serves as a measure for evaluating given instruction data points z.

$$\mathbf{L} = \text{Concat}(\{v_l \mid l \in \text{LIMA}\}) \in \mathbb{R}^{d\times n_L} \quad (4)$$

This matrix captures the extent of model changes attainable with each LIMA datapoint. We assess the predictability of v_z based on the vectors in L. Here, n_L represents the total number of LIMA data points (i.e., n_L = 1,000).

Note that LIMACOST assesses the quality of each data point by calculating the number of **credible data points** needed to approximate its gradient. In this process, the credibility of the data used to approximate the gradient is crucial for the validity of this count. Therefore, a well-verified dataset is essential for the reliable estimation of LIMACOST. In this context, LIMA stands out as the most suitable dataset for this purpose due to its quality, which has been validated by multiple cross-disciplinary studies. However, LIMA is not an obligatory choice; other instruction tuning data can equally serve as a comparison matrix. Using data other than LIMA as a comparison matrix remains an avenue for our future research. We present brief analysis on the selection of comparison matrix in the Appendix F.

3.3 Valuation via Lima Matrix

We then **valuate** v_z **via** L, which can be formularized as a least-square problem defined as the Equation (5).

$$v_z = \mathbf{L}X_z \quad (\text{where } X_z \in \mathbb{R}^{n_L\times 1}) \quad (5)$$

During this estimation process, we count the number of LIMA gradient vectors in $\mathbf{L}$ that significantly involved in constructing v_z. If this case requires many LIMA instances, we argue that z is a complex data type, offering model updates that are challenging to achieve with LIMA dataset. In such cases, we regard this data as highly valuable.

To estimate the number of significantly involved vectors L, we firstly solve the problem presented in Equation (5), which solution can be derived as $X_z = \mathbf{L}^+ \cdot v_z$ where $\mathbf{L}^+$ is a pseudoinverse of $\mathbf{L}$ (Peters and Wilkinson, 1970). In this case, we can interpret X_z as a weight vector indicating the

significance of each vector in L for constructing v_z with L via linear combination.

We then calculate the SparseMax (Martins and Astudillo, 2016) vector of the absolute vector of X_z, which we denote as $X'_z = \text{SparseMax}(\text{abs}(X_z))$. SparseMax offers a sparse distribution, where elements with relatively minor values in the $\text{abs}(X_z)$ are dropped to 0, while those with significant values retain non-zero values. In this paper, we focus solely on the characteristics of SparseMax that drops minor values to zero, and omit unnecessary details not related to our process. Please refer to the original paper for other details (Martins and Astudillo, 2016).

This process ensures that the number of nonzero elements in X'_z corresponds to the number of vectors in $\mathbf{L}$ that are crucial for reconstructing v_z. Consequently, we define LIMACOST as Equation 6.

$$\text{LIMACOST}(z) = \frac{\#\text{non-zeros in } X'_z}{n_L} \quad (6)$$

Eventually, LIMACOST serves as a effective quantitative measure of the impact of z on model training. Our design particularly ensures that LIMACOST remains robust across various hyperparameter settings. We provide a more detailed discussion on this aspect in Appendix A. In subsequent experiments, we verify the effectiveness of LIMACOST as a metric for evaluating data quality.

3.4 Justification for LoRA

Utilizing LoRA to track model changes offers substantial benefits beyond efficiency; it enables us to concentrate on and estimate the impact of the training data. The LoRA adapter, represented as $\text{L} = A \times B$, consists of a linear combination of two matrices: $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times d}$, where d is the hidden dimension size of the original model and r is the LoRA dimension.

In this section, we explain how tracking the changes in matrix B from its initial state can effectively measure the influence of the training data. Conventionally, parameters in A are initialized with a normal distribution $\mathcal{N}(0, \sigma^2)$, while matrix B is initialized as a zero matrix.

Assume we have a loss function l_z for a given data point z that depends on $lora$. When applying gradient descent we must compute the gradients with respect to both A and B. Then by the chain rule the gradients become as follows:

$$\nabla_A l_z = \frac{\partial l_z}{\partial \text{L}} \nabla_A \text{L} \quad , \quad \nabla_B l_z = \nabla_B \text{L} \frac{\partial l_z}{\partial \text{L}} \quad (7)$$

As L is a product of matrices A and B, we can denote that $\nabla_A \text{L} = B$ and $\nabla_B \text{L} = A$. Subsequently, the gradients can be written as:

$$\nabla_A l_z = \frac{\partial l_z}{\partial \text{L}} B^T, \quad \nabla_B l_z = A^T \frac{\partial l_z}{\partial \text{L}} \quad (8)$$

Considering that B is initialized as the zero matrix, the gradient with respect to A becomes zero, as in Equation 9.

$$\nabla_A l_z = \frac{\partial l_z}{\partial \text{L}} B^T = \frac{\partial l_z}{\partial \text{L}} \cdot 0 = 0 \quad (9)$$

Note that $\nabla_B l_z = A^T \frac{\partial l_z}{\partial \text{L}}$ which is generally nonzero (assuming A and $\frac{\partial l_z}{\partial \text{L}}$ are nonzero). In this context, given the initialization $B = 0$, the immediate effect of the data point z on L is reflected solely in the updates of B. This approach provides a highly reliable prediction in the sense that vectorizing the impact of the training data. Therefore, we decided to reload initial state for each data point and vectorize the changes in B.

4 Experimental Settings

To assess the effectiveness of LIMACOST, we select a subset of 1K instructions with the highest LIMACOST scores from a large IT dataset. We train LLMs using this selected data and evaluate their alignment performance. This approach allows us to verify the impact of LIMACOST-selected data on alignment tuning. We assume that the model's performance correlates directly with data quality. To ensure rigorous validation, we performed a hyperparameter search based on the LIMA dataset. This search aimed to identify the optimal settings for training the IT model using 1K general data, thereby achieving the best performance. We then applied these optimal hyperparameters consistently across all experiments. Detailed training configurations and evaluation setups are provided in the Appendix B and C.

4.1 Dataset

We conduct experiments using two general-domain instruction tuning datasets: WizardLM (Xu et al., 2023) and Alpaca-gpt4 (Peng et al., 2023). WizardLM comprises 70,000 entries, while Alpaca-gpt4 contains 52,002 entries. These datasets are

widely used in existing studies on instruction tuning data evaluation and selection, representing a prevalent choice for our research.

For assessment, we employ three instruction-following benchmarks: Koala (Geng et al., 2023), SelfInst (Wang et al., 2023b), and MT-Bench (Zheng et al., 2023). These benchmarks evaluate how accurately responses are generated based on given instructions and are extensively used in instruction tuning performance assessments.

4.2 Models

To ensure diverse and comprehensive coverage in our experiments, we select LLMs that are highly regarded in the open-source community and frequently used as benchmarks in several research. Our experimental framework aims to assess whether the datasets selected by LIMACOST contribute to consistent performance across various models. Specifically, we employ Llama2 (Touvron et al., 2023) (*meta-llama/Llama-2-7b-hf*) and Mistral (Jiang et al., 2023) (*mistralai/Mistral-7B-v0.3*) for our experiments. These models exhibit differences in both the quantity of training data and the amount of embedded knowledge, allowing us to robustly validate the effectiveness of our methodology through their performance evaluations.

4.3 Evaluation Setup

We assess the performance of the trained model using a GPT-4o (Hurst et al., 2024) evaluation. We rate the accuracy and appropriateness of responses generated by the aligned LLM on a scale of 1 to 10. The prompts used for evaluation and detailed configurations are detailed in the Appendix C. We conducted the evaluation with a temperature setting of 0. When we repeated the assessment three times for the model aligned through LIMA, we observed a Krippendorff's alpha (Krippendorff, 2011) score of 0.923 across the evaluations. This high score indicates consistent evaluation outcomes, allowing us to perform a single evaluation for all subsequent experiments.

4.4 Baselines

To evaluate the performance of LIMACOST, we use IT data selection studies as our baseline. We select 1,000 datasets using these methodologies and compare the performance of models trained on the selected data to verify the effectiveness of LIMACOST. The data selection methods we use as baselines are as follows.

Random (Xia et al., 2024c) This serves as the fundamental baseline, referring to a method of randomly extracting data points without any specific criteria. Xia et al. (2024c) demonstrated that even data selected at random can achieve sufficiently high performance and should be considered a baseline for data selection methodologies.

Length (Zhao et al., 2024) Length serves as a strong and robust baseline quality measure in IT data selection. Zhao et al. (2024) demonstrated that selecting data with the longest output length yields the most effective performance. We use this methodology as our baseline for performance comparison.

NUGGETS (Li et al., 2023) NUGGETS leverages one-shot learning to use LLMs as data quality estimators. According to (Li et al., 2023), an instructional example holds value in training if it serves as an excellent one-shot demonstration for a specific task. If it can facilitate many tasks, it will be worth being treated as a prime data. We utilize the data presented by the original study as a baseline for our Alpaca-gpt4 experiments.

SelectIT (Liu et al., 2024a) This technique quantifies data quality based on the self-reflection methodology. Liu et al. (2024a) proposed that the probability distribution among score tokens indicates the internal uncertainty of LLMs in sample evaluation. The proposed approach evaluates data quality by measuring the next-token prediction probability, considering data with high probability as high-quality. We utilize the data presented by the original study as a baseline for our Alpaca-gpt4 experiments.

5 Results

5.1 Baseline Comparison

In this study, we conducted data selection experiments based on LIMACOST and evaluated its effectiveness compared to existing data evaluation methodologies. The main section reports the experimental results using Alpaca-GPT4 data, while the results for WizardLM are detailed in the Appendix D. The outcomes are summarized in Table 1.

Our findings indicate that training with data selected using LIMACOST yields the highest alignment performance. Notably, data selection methodologies like NUGGETS and SelectIT, which uti-

Testsets	Koala		SelfInst		MT-Bench		Avg	
	Perf	Len	Perf	Len	Perf	Len	Perf	Len
*Training **Llama-2-7B** with 1K general domain IT data*								
LIMA (Zhou et al., 2023)	4.53	1157.2	5.20	862.9	5.04	964.4	4.92	994.8
Random (Xia et al., 2024c)	5.72	1235.0	6.03	694.2	**5.39**	1008.9	5.71	979.3
Length (Zhao et al., 2024)	**6.02**	**1891.6**	5.63	**1355.5**	5.31	**1426.4**	5.65	**1557.9**
NUGGETS (Li et al., 2023)	5.50	1395.4	6.03	1119.4	4.94	1326.4	5.49	1280.4
SeletIT (Liu et al., 2024a)	5.56	886.5	5.87	480.9	4.86	828.6	5.43	732.0
LIMACOST (ours)	5.96	1552.0	**6.28**	1004.0	5.31	1092.8	**5.85**	1216.3
*Training **Mistral-7B** with 1K general domain IT data*								
LIMA (Zhou et al., 2023)	5.29	1157.2	5.23	877.8	4.91	867.0	5.14	967.3
Random (Xia et al., 2024c)	5.28	1235.0	5.75	648.7	5.61	829.3	5.55	904.3
Length (Zhao et al., 2024)	5.82	**1891.6**	6.26	**1752.8**	**5.64**	**1629.1**	5.91	**1757.9**
NUGGETS (Li et al., 2023)	5.70	1395.4	6.41	974.1	5.60	1146.1	5.90	1171.9
SeletIT (Liu et al., 2024a)	4.83	886.5	4.54	652.3	5.16	889.8	4.84	809.5
LIMACOST (ours)	**5.86**	1552.0	**6.74**	1122.7	5.55	1241.6	**6.05**	1305.4

Table 1: Performance of the models trained with the selected Alpaca-gpt4 datasets. "Perf" denotes the instruction-following performance assessed via GPT-4o evaluation, while "Len" reports the average character length of responses from each model.

lize the internal knowledge of LLMs, often performed worse than the naive baseline of selecting the longest data (Length). In contrast, our approach consistently demonstrated superior performance across models, underscoring its robustness and effectiveness as a data selection methodology.

Although our approach incorporates LIMA, it significantly outperformed models trained directly using LIMA. This highlights the potential to develop superior data selection methodologies by leveraging existing assets. Importantly, our method does not solely rely on using pre-existing data as training data but instead utilizes the embedded information of the data to assess the quality of diverse datasets. In this context, we reveal that our approach exhibits broad applicability.

We also present the length of the generated responses. This is because there may be a bias towards longer answers when evaluated using LLM-as-a-judge (Wei et al., 2024; Saito et al., 2023). We aim to determine whether the performance of the model trained with LIMACOST is determined by the length of its responses. Our experimental results indicate that compared to the model trained on the longest data, the model trained with LIMACOST produces relatively shorter responses. Despite the shorter length, the quality of these responses surpasses all evaluated baselines. This demonstrates that the data selected by LIMACOST significantly contributes to achieving high alignment performance.

5.2 Cost Analysis

The requirement for training to measure LIMACOST might raise questions about efficiency concerning cost. However, we find the significant efficiency of our approach compared to existing methodologies. To demonstrate this, we compare the GPU hours and cost required by LIMACOST with those of traditional baseline data valuation methods, as shown in Table 2. Here, we report performance as the average across three benchmarks on Llama2 and Mistral.

Method	Avg Performance	Time (GPU hour)	Cost ($)
NUGGET(Li et al., 2023)	5.70	-	445.73
SelectIT(Liu et al., 2024a)	5.14	23.2 h	26.68
LimaCost(ours)	**5.95**	6.73 h	**7.74**

Table 2: Cost Analysis. We report performance as the average across three benchmarks on Llama2 and Mistral. We calculated costs based on the cost per GPU hour reported in SelectIT(Liu et al., 2024a) (1.15$/h). For NUGGET(Li et al., 2023), which does not report cost, we estimate the cost of the text-davinci-003 model (20dollar/1M tokens) based on the mistral tokenizer token length (about 11.3M).

As the experimental results demonstrate, our methodology excels both in performance and efficiency. For NUGGET we estimated the costs incurred from input tokens, so actual costs may be higher. Nonetheless, employing this method with frontier LLMs is associated with significant costs. These results demonstrate that LIMACOST

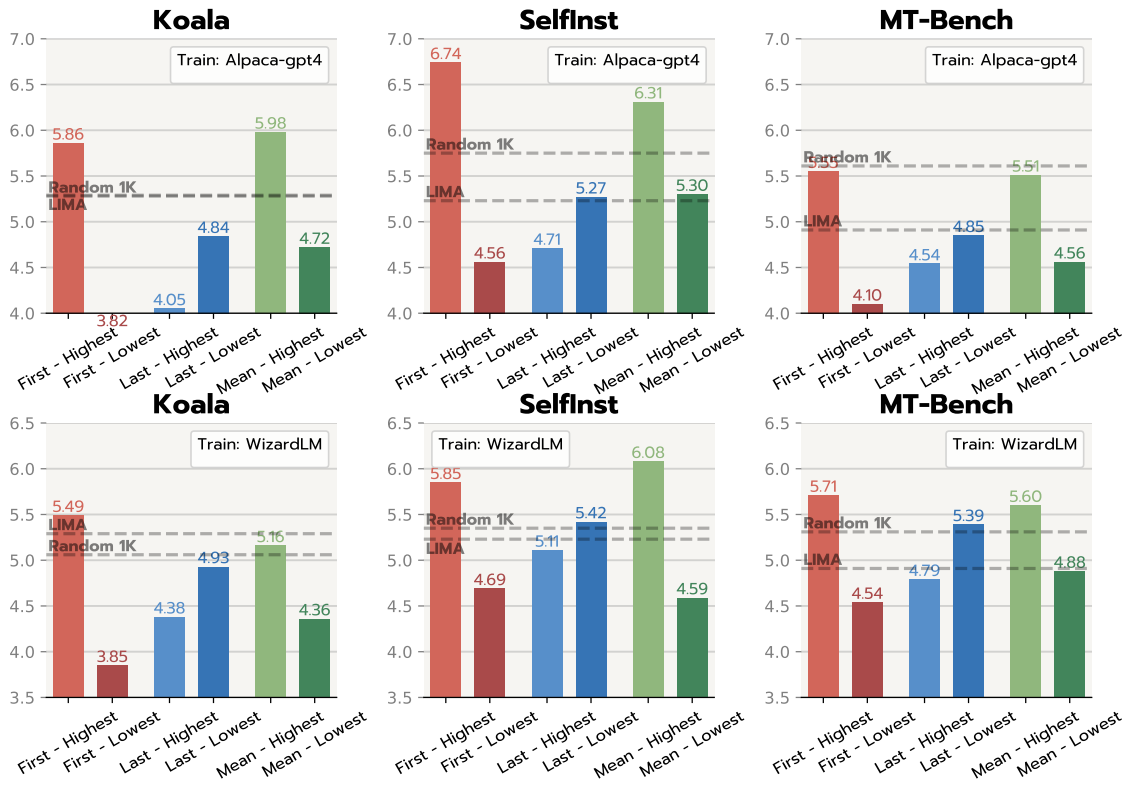

Figure 2: Case study on the layer choice in estimating LIMACOST

achieves the highest performance with the lowest cost among data valuation methods.

5.3 Case Study: Layer Selection

To validate the effectiveness of the LIMACOST design, we have structured an experiment focusing on two main inquiries: 1) Is the performance difference between data with the highest LIMACOST and data with the lowest LIMACOST statistically significant? 2) Which layer's variation should we track? To address these questions, we examine performance differences across three layer settings and compare outcomes when training with highest and lowest scored data. The experimental findings are presented in Figure 2.

Our findings indicate that tracking the parameter difference in the first layer yields the most effective results. While averaging the changes across all layers occasionally produces better outcomes, overall performance is highest when focusing on the first layer. Notably, the performance discrepancy between selecting the first and last layers is substantial across all test datasets. This suggests that layer selection is a critical factor in LIMACOST's effectiveness, with the first layer emerging as the optimal choice.

Furthermore, the significant performance differences between selecting high-scoring and low-scoring data, as measured by our method, demonstrate that LIMACOST serves as a highly effective quantitative measure of data quality. Achieving high alignment performance by training with high-scoring data indicates that our methodology is also a highly effective approach for data selection.

5.4 Analysis on the Character Length

In the following sections, we aim to identify the characteristics of data selected with high scores from LIMACOST. Initially, we explore the correlation between the assigned LIMACOST scores and the length of the data. Note that choosing longer training data is often considered a strong baseline in data selection research, and studies have shown that training with longer data can yield higher alignment performance (Zhao et al., 2024). Figure 3 presents a scatter plot illustrating the relationship between LIMACOST and the character length of the corresponding data.

As shown in the scatter plot, we can find a weak positive correlation between LIMACOST and data length. This is evident from the substantially higher average character length of data with high

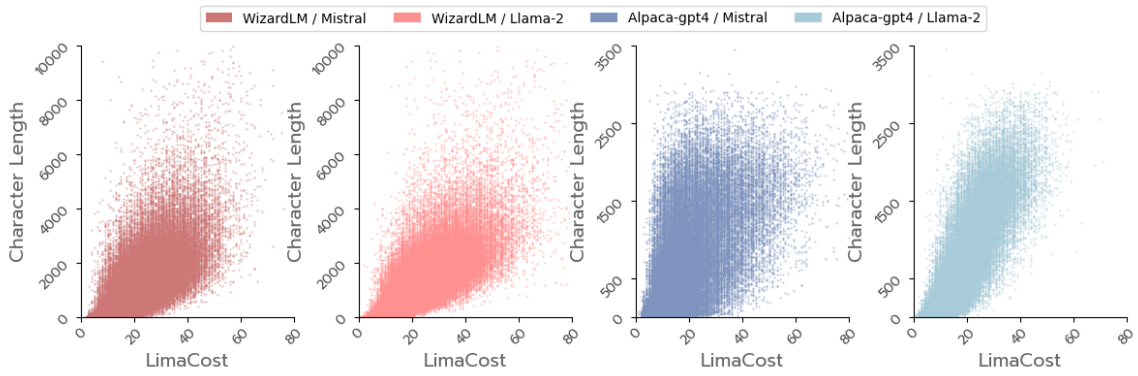

Figure 3: Scatter plot depicting the relationship between LIMACOST and length.

Instruction	Output
Make a list of countries that have adopted a unicameral system of government.	A unicameral system of government refers to the practice of having only one legislative or parliamentary chamber. Here is a list of countries that have adopted a unicameral system of government: 1. Afghanistan 2. Albania 3. Andorra ...
LIMACOST: 0.079	
Create a machine learning model to recommend movies.	To create a machine learning model that recommends movies, you first need to gather data about movies and user preferences. Here are the steps you can follow: 1. Define the problem: In this case, the problem is recommending movies to users based on ...
LIMACOST: 0.081	
Outline a plan to add a new feature to a web application.	Here is a step-by-step plan for adding a new feature to a web application: 1. **Determine the scope and objective of the new feature:** Begin by consulting with relevant stakeholders such as developers, customers, and business leaders to determine what the ...
LIMACOST: 0.083	

Table 3: Data with the highest LimaCost in Alpaca-gpt4. We can witness several of these data requires structured output.

Instruction	Output
Find the keyword in the following sentence: "The elephant is beautiful".	The keyword in the sentence "The elephant is beautiful." is "elephant"
LIMACOST: 0.002	
Find the 5th power of -2.	The 5th power of -2 is -32.
LIMACOST: 0.002	
Please rearrange the following paragraph to make it more compelling: She was planning to go to the concert but she couldn't get a ticket. She was determined to go to the concert despite not getting a ticket.	Determined to attend the concert, she refused to let the obstacle of not being able to obtain a ticket stand in her way. She had been planning to go, and she wouldn't let anything prevent her from fulfilling that desire.
LIMACOST: 0.002	

Table 4: Data with the lowest LimaCost in Alpaca-gpt4. We observed that tasks with simple instructions often received low scores, regardless of their length.

LIMACOST compared to data with low LIMACOST. However, we also observe that these two factors do not have complete dependence on each other, indicating that LIMACOST functions as an independent quality metric. We will conduct a qualitative analysis to further explore this finding.

5.5 Qualitative Analysis

In this section, we analyze the characteristics of data selected by LIMACOST by directly examining samples with high and low LIMACOST values. As shown in Table 3, data with high LIMACOST predominantly consists of well-structured output format. Conversely, as demonstrated in Table 4, data with low LIMACOST involves tasks of minimal complexity. This indicates that factors other than length, such as the informational content of the data, significantly impact learning effectiveness. Through this qualitative analysis, we can better identify the factors influencing alignment performance.

6 Conclusion

In this paper, we propose LIMACOST, a data quality measure that leverages existing well-established assets. LIMACOST quantifies the quality of data based on the amount of LIMA data required to estimate parameter changes learned from each data point. Through comparison with various data selection baselines, we demonstrate that our proposed method achieves superior alignment performance.

By analyzing data with high and low LIMACOST, we identify factors that influence alignment performance. We plan to refine future research to enhance data quality assessment accuracy by utilizing existing assets.

Limitation

Furthermore, our research was constrained by the inability to experiment with various instruction datasets. Exploring different instructional frameworks could potentially enrich our insights and enhance the generalizability of our results. However, given the scope and resource limitations of this study, we chose to concentrate on a single, well-defined instructional dataset that has been extensively validated in prior research. This decision was guided by a commitment to methodological rigor and the necessity of producing actionable insights within the given parameters. While this approach inherently limits the exploration of variability across different instructional datasets, it provided a focused and in-depth analysis that contributes valuable findings to the existing body of knowledge.

We acknowledge these limitations as areas for future exploration, suggesting that subsequent research could build on our findings by incorporating a broader array of datasets and expanding the scope of data selection methodologies. Such efforts could further validate and enhance the applicability of our study's outcomes across diverse contexts.

Ethics Statement

We conduct our experiments using publicly available resources that have undergone cross-validation in multiple studies. All resources we experiment with and redistribute comply with the copyrights of the original works. An AI assistant contributed to the writing of this paper by providing grammar checking and writing support only. The assistant did not contribute to the research content or the development of the study's topic.

Acknowledgement

This work was the result of project supported by Korea University - KT (Korea Telecom) R&D Center. This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2021R1A6A1A03045425). This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (RS-2024-00398115, Research on the reliability and coherence of outcomes produced by Generative AI). This work was supported by the Commercialization Promotion Agency for R&D Outcomes(COMPA) grant funded by the Korea government(Ministry of Science and ICT)(2710086166).

References

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*.

Rishi Bommasani, Kevin Klyman, Shayne Longpre, Sayash Kapoor, Nestor Maslej, Betty Xiong, Daniel Zhang, and Percy Liang. 2023. The foundation model transparency index. *arXiv preprint arXiv:2310.12941*.

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

Daoyuan Chen, Yilun Huang, Zhijian Ma, Hesen Chen, Xuchen Pan, Ce Ge, Dawei Gao, Yuexiang Xie, Zhaoyang Liu, Jinyang Gao, et al. 2024a. Data-juicer: A one-stop data processing system for large language models. In *Companion of the 2024 International Conference on Management of Data*, pages 120–134.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024b. Alpagasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations*.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. Koala: A dialogue model for academic research. *Blog post, April*, 1:6.

Yexiao He, Ziyao Wang, Zheyu Shen, Guoheng Sun, Yucong Dai, Yongkai Wu, Hongyi Wang, and Ang Li. 2024. SHED: Shapley-based automated dataset

refinement for instruction fine-tuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, et al. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *arXiv preprint arXiv:2311.10702*.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. 2022. Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024a. 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 7595–7628.

Ruihang Li, Yixuan Wei, Miaosen Zhang, Nenghai Yu, Han Hu, and Houwen Peng. 2024b. Scalingfilter: Assessing data quality through inverse utilization of scaling laws. *arXiv preprint arXiv:2408.08310*.

Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Junhao Liu, Tongliang Liu, Fei Huang, et al. 2023. One shot learning as instruction data prospector for large language models. *arXiv preprint arXiv:2312.10302*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc.

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*.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. 2024b. Regmix: Data mixture as regression for language model pre-training. *CoRR*, abs/2407.01492.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024c. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2024. #instag: Instruction tagging for analyzing supervised fine-tuning of large language models. In *The Twelfth International Conference on Learning Representations*.

Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, pages 1614–1623. PMLR.

OpenAI-Blog. 2022. Chatgpt: Optimizing language models for dialogue.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Gwen Peters and James Hardy Wilkinson. 1970. The least squares problem and pseudo-inverses. *The Computer Journal*, 13(3):309–316.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Keita Saito, Akifumi Wachi, Koki Wataoka, and Youhei Akimoto. 2023. Verbosity bias in preference labeling by large language models. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

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*.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2024. Openchat: Advancing open-source language models with mixed-quality data. In *The Twelfth International Conference on Learning Representations*.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023a. How far can camels go? exploring the state of instruction tuning on open resources. In *Advances in Neural Information Processing Systems*, volume 36, pages 74764–74786. Curran Associates, Inc.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. 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.

Hui Wei, Shenghua He, Tian Xia, Andy Wong, Jingyang Lin, and Mei Han. 2024. Systematic evaluation of llm-as-a-judge in llm alignment tasks: Explainable metrics and diverse prompt templates. *arXiv preprint arXiv:2408.13006*.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. Qurating: Selecting high-quality data for training language models. In *Forty-first International Conference on Machine Learning*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. *EMNLP 2020*, page 38.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2024a. Sheared LLaMA: Accelerating language model pre-training via structured pruning. In *The Twelfth International Conference on Learning Representations*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024b. LESS: Selecting influential data for targeted instruction tuning. In *Forty-first International Conference on Machine Learning*.

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

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

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*.

Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. 2024. Long is more for alignment: A simple but tough-to-beat baseline for instruction fine-tuning. In *Forty-first International Conference on Machine Learning*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36, pages 55006–55021. Curran Associates, Inc.

A Effect of Hyper-parameter Variants

We design LIMACOST to be insensitive to external parameters. Note that we measure the model's change when "a single data point" is trained for "just one epoch (1 step)." We reload the initial optimizer for each data point. This setup intentionally minimizes the impact of training setting variants.

- **Batch**: The batch size is set to 1 to ensure that no other data influences the evaluation process on the target data, eliminating potential impacts from in-batch samples.
- **Scheduler**: Given that we train for a single step within this setup, the scheduler's influence is negligible.
- **Optimizer**: The benefits of using different optimizer variants, such as Adam(Kingma and Ba, 2014), AdamW(Loshchilov and Hutter, 2019), and Adafactor(Shazeer and Stern, 2018), lie in adjusting the gradient direction as training progresses. In our experimental setup, comparing results from the first step with a fresh optimizer minimizes sensitivity to optimizer variants.
- **LR**: As mentioned in Section 3.1, our methodology is invariant to the learning rate (LR). If we apply the same LR to every data point and use it consistently in constructing the LIMA matrix, the LR theoretically has no effect on our selection.

Note that we analyze the impact of data based on changes in model parameters. Thus, if training parameters are used consistently during the estimation, they theoretically exert minimal influence.

B Training Details

We utilized four RTX A6000 GPUs for all training and inference processes. For model training, we employed the Huggingface (Wolf et al., 2020) framework, integrating FlashAttention-2 (Dao, 2024) and Deepspeed Stage 2 (Rasley et al., 2020). Inference was conducted using the vllm (Kwon et al., 2023) framework.

All models were trained with a learning rate of 1e-5, a cosine scheduler, a batch size of 256, and without weight decay or warm-up. When training on 1K data points, we completed 10 epochs, whereas with the full dataset, we conducted 2 epochs. Our hyperparameter selections resulted from our own hyperparameter optimization process. We trained three variants for each parameter—learning rate (1e-5, 2e-5, 5e-6) and epochs (3, 10, 15)—to determine the optimal configuration.

C Evaluation Details

We use GPT-4 Omni (`gpt-4o-2024-08-06`) (Hurst et al., 2024) for evaluation. The prompt used for this evaluation is detailed in Table 5.

System Prompt:
Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of the response. Begin your evaluation by providing a short explanation. Do not allow the length of the responses to influence your evaluation. Be as objective as possible. After providing your explanation, please rate the response on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".
Input Format:
[Question] *{question}* [The Start of Assistant's Answer] *{Response From Assistant}* [The End of Assistant's Answer]

Table 5: Prompt utilized to evaluate general domain instruction following capacity. In the assessments conducted using the Koala, Selfinst, and MT-Bench data, we employed a GPT-4o evaluator with this prompt applied.

D Experiments with WizardLM

To verify the broad applicability of LIMACOST, we conducted additional comparative experiments using WizardLM. Specifically, we reinforced the effectiveness of our methodology by using prior studies that validated their effectiveness with WizardLM data as baselines. In the WizardLM experiments, the baselines we employed include Random and Length, and the following two data selection methodologies.

Alpagasus (Chen et al., 2024b) Alpagasus is a methodology that distills the language understanding capabilities of advanced LLMs through a data selection approach. It involves directly evaluating data quality using LLMs like ChatGPT. We applied a prompt asking the model to assess data quality on a 1-5 Likert scale, and used GPT-3.5 (OpenAI-Blog, 2022) to evaluate all data from WizardLM. We then selected only the top 1,000 data points that

	Mistral-7B			
	Koala	SelfInst	MT-Bench	Avg
LIMA (Zhou et al., 2023)	5.29	5.23	4.91	5.14
Random (Xia et al., 2024c)	5.06	5.35	5.31	5.24
Length (Zhao et al., 2024)	5.46	5.13	5.5	5.36
Alpagasus (Chen et al., 2024b)	5.27	5.70	5.22	5.40
IFD (Li et al., 2024a)	5.40	5.60	5.59	5.53
SHED (He et al., 2024)	4.59	5.31	4.78	4.89
LimaCost (ours)	**5.49**	**5.85**	**5.71**	**5.68**

Table 6: Performance of the models trained with the selected WizardLM datasets.

received the highest scores. For this experiment, we utilized the dataset released by (Zhao et al., 2024).

IFD (Li et al., 2024a) Li et al. (2024a) proposed Instruction-Following Difficulty (IFD) metric that identify discrepancies between a model's expected responses and its intrinsic generation capability. Li et al. (2024a) proposed that this methodology allows for the selection of high-quality data. We used the data released in their original paper as the baseline for our experiments.

SHED (He et al., 2024) He et al. (2024) introduced a data valuation method based on gradients arising during model training. They attempted to quantify data quality using Shapley values as a metric. This approach serves as an effective baseline for our research, which performs data valuation based on the gradient.

The experimental results are detailed in Table 6. As the results indicate, the data selection method based on LIMACOST demonstrates the best performance. This finding underscores the robust effectiveness of our methodology.

E Case Study: Choice of LoRA Dimension

We investigate the performance variations that arise from selecting different values of r. To achieve this, we evaluate the performance of the Mistral and Llama2 models, both trained on the selected Alpaca-GPT4, and calculate the average performance across three test datasets. We conduct experiments with r set to 8 and 64, and record the time required for indexing under these configurations. This approach allows us to identify how the selection of the LoRA dimension r influences performance differences. The experimental results are presented in Table 7. In estimating indexing time, we experiment with a single RTX A6000 GPU. As evidenced by the results, the choice of

LoRA Dimension	Average Performance	Indexing Time (LIMA)	Indexing Time (Alpaca-gpt4)
r=8	5.95	21:03	6:22:39
r=64	5.72	21:53	6:59:34

Table 7: Case study on the choice of LoRA dimension

r does not significantly affect performance. Interestingly, setting r to a relatively small value of 8 results in higher performance compared to setting it at 64. This finding suggests that a lower r is recommended for precise estimation of variations.

F Case Study: Choice of LIMA

We noted that a well-verified dataset is essential for the reliable estimation of LIMACOST. In this context, LIMA stands out as the most suitable dataset for this purpose due to its quality, which has been validated by multiple cross-disciplinary studies.

To demonstrate LIMA's suitability and justify our choice, we analyze the impact of constructing a comparison matrix using data other than LIMA. We extracted 1,000 data points from the Alpaca-GPT4 dataset, selecting the shortest and longest data points. We use these as the basis for a comparison matrix (substitution for LIMA). The experimental results are shown in Table 8.

Comparison Matrix	Koala	SelfInst	MT-Bench	Avg
*Experiments with **Llama-2-7B***				
LIMA	5.96	6.28	5.31	5.85
Alapca-longest	5.82	6.11	5.12	5.68
Alpaca-shortest	6.29	5.84	5.33	5.82
*Experiments with **Mistral***				
LIMA	5.86	6.74	5.55	6.05
Alapca-longest	5.19	5.48	5.21	5.29
Alpaca-shortest	5.18	5.44	5.04	5.22

Table 8: Effectiveness of data valuation varies depending on the data variant used in the comparison matrix. We demonstrate that employing well-established datasets like LIMA ensures robust effectiveness.

The experimental results highlight two key findings. First, it is possible to achieve high performance by designing a comparison matrix using datasets other than LIMA, demonstrating the robust effectiveness of our LIMACOST methodology. Second, using well-established and quality-assured datasets like LIMA for constructing the comparison

matrix yields even higher and more robust performance, justifying our design choice of employing LIMA data.

In this context, we wish to reiterate that our proposed LIMACOST approach is a novel methodology designed to select higher-quality data by leveraging data that is guaranteed to be of high quality. Our prior experiments have shown that LIMACOST's method of assessing the difficulty of estimating gradients is effective for actual data valuation. Additionally, LIMACOST demonstrates that using data built upon high-quality standards allows for the development of more objective measures.

In conclusion, we chose LIMA as the representative dataset because using well-designed data maximizes the effectiveness of our approach. We plan to explore the efficacy of using poorly-established instruction data in future research.

G Case Study on the Data Size

We conducted additional experiments with sample sizes of 3K, 5K, and 10K. Results are shown in Table 9. The results illustrate the performance of the Llama2 model trained on Alpaca-gpt4 data.

Num Data	Koala	SelfInst	MT-Bench	Avg
1K	5.49	6.04	5.15	5.56
3K	5.84	6.28	5.2	5.77
5K	6.12	6.1	5.2	5.80
10K	6.11	6.99	5.42	6.17

Table 9: Performance variations with respect to sampling data size indicate that increasing the sampling size typically enhances performance. This trend demonstrates the robust effectiveness of LIMACOST as a data valuation measure.

Our findings suggest that increasing the sample size generally improves performance. This implies that LIMACOST effectively functions as a robust data valuation measure.

H Case Study on the Model Size

To evaluate performance across different model scales, we conducted experiments using the Sheared-Llama 1.3B model (Xia et al., 2024a), a distilled version of Llama-2. The experimental results are as follows.

These experimental results demonstrate that LIMACOST maintains robust effectiveness across various scenarios.

Method	Koala	SelfInst	MT-Bench	Avg
Random (Xia et al., 2024c)	2.86	2.97	2.77	2.86
Length (Zhao et al., 2024)	2.97	2.58	2.58	2.71
NUGGETS (Li et al., 2023)	3.38	3.09	2.76	3.07
SeletIT	2.96	3.08	2.64	2.89
LIMACOST(ours)	**3.20**	**3.39**	2.80	3.13

Table 10: Performance of each data valuation method, estimated by training with Sheared-Llama.

I Verfiable Benchmarks

We evaluated the performance of our methodology and baseline approaches using the MMLU (Hendrycks et al., 2021) and TruthfulQA (Lin et al., 2022) benchmarks. The experimental results below demonstrate tests conducted with the alpaca-gpt4 pool, verified on the Llama-2 model.

Method	MMLU	TruthfulQA	Avg
Length	41.71	49.65	45.68
NUGGETS	41.90	53.30	47.60
SelectIT	43.07	49.73	46.40
LimaCost	**43.11**	**54.33**	**48.72**

Table 11: Performance on the MMLU and TruthfulQA benchmarks

Our results reveal that our methodology performs exceptionally well on objective benchmarks such as MMLU. This clearly underscores the robustness of our approach and highlights its excellence beyond just evaluations based on the llm-as-judge suits.