Rethinking Data Selection at Scale: Random Selection is Almost All You Need

Anonymous ACL submission

Abstract

Supervised fine-tuning (SFT) is crucial for aligning Large Language Models (LLMs) with human instructions. The primary goal during SFT is to select a small yet representative subset of training data from the larger pool, such that fine-tuning with this subset achieves results comparable to or even exceeding those obtained using the entire dataset. However, most existing data selection techniques are designed for small-scale data pools, which fail to meet the demands of real-world SFT scenarios. In this paper, we replicated several self-scoring methods—those that do not rely on external model assistance—on two million-scale datasets, and found that nearly all methods struggled to significantly outperform random selection when dealing with such large-scale data pools. Moreover, our comparisons suggest that, during SFT, diversity in data selection is more critical than simply focusing on high-quality data. We also analyzed the limitations of several current approaches, explaining why they perform poorly on large-scale datasets and why they are unsuitable for such contexts. Finally, we found that filtering data by token length offers a stable and efficient method for improving results. This approach, particularly when training on long-text data, proves highly beneficial for relatively weaker base models, such as Llama3.

1 Introduction

With the advent of large language models (LLMs) such as ChatGPT, we have observed significant advancements in tasks involving instruction following (Wang et al., 2023b), intent comprehension (Lu et al., 2023), and text generation (Zhao et al., 2023). One of the primary objectives of developing LLMs is to harness their potential for generalizing to unseen natural language processing (NLP) tasks. To achieve this aim, many LLMs focus on precisely aligning with human instructions.

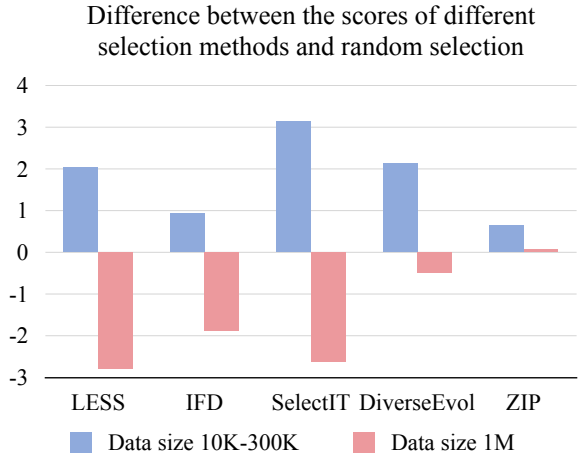

Figure 1: The discrepancy between each methods and random selection on BBH benchmark (Suzgun et al., 2022). The Y-axis represents the differential score, which is computed by subtracting the random selection score from the scores obtained using various methods.

Recent studies indicate that supervised fine-tuning (SFT) can customize LLMs for specific domains, tasks, or applications by utilizing well-crafted data. According to the study in (Zhou et al., 2024a), it is feasible to fine-tune a pre-trained language model with a relatively small set of examples. Building on this insight, several papers have explored data selection strategies for SFT of LLMs (Wang et al., 2024; Qin et al., 2024), emphasizing the importance of enhancing the quality of instruction tuning (IT) data or increasing data diversity. These strategies can be classified into two primary categories: (1) Extenral-scoring methods, which require support from more sophisticated external models like GPT-4 to score the data for the subsequent selection (Lu et al., 2023; Chen et al., 2023; Du et al., 2023; Liu et al., 2023; Zhou et al., 2024b); (2) Self-scoring methods, which leverage LLMs themselves as data scorers (Li et al., 2023d,b; Liu et al., 2024; Xia et al., 2024; Yin et al., 2024).

Existing SFT data selection methods, both external-scoring and self-scoring, are evaluated using well-known IT datasets like alpaca-GPT4 (Peng

et al., 2023), Dolly (Conover et al., 2023), FLAN (Longpre et al., 2023), WizardLM (Xu et al., 2024), and ShareGPT (Chiang et al., 2023). These datasets are small and come from a single source. However, during SFT, much larger data volumes, usually in the hundreds of thousands to millions, are often needed. For instance, Qwen2 (qwe, 2024) used over 500,000 data samples in SFT. Thus, for effective LLM utilization, large-scale instruction-following data is crucial in SFT. Furthermore, large-scale data should not only be abundant but also diversified, including professional annotations, real user data, or model-generated data, across various types like code, math, conversations, and knowledge Q&A. This disparity highlights a gap between current SFT data selection and real-world applications. To study how dataset size impacts selection strategy performance, we compared outcomes from existing methods with random selection within datasets ranging from 10K-30K to 1M on Llama3-8B (AI@Meta, 2024). Figure 1 shows that as dataset size grows to 1M, these methods perform worse compared to random selection. "Data size 10K-300K" refers to sources from original method papers. "Data size 1M" refers to the Openhermes2.5-1M dataset (Teknium, 2023).

Motivated by this discovery, we reconsider whether SFT data selection methods are viable for large-scale IT datasets. Given the high costs of external-scoring techniques (Liu et al., 2023), we focus on self-scoring methods. Referring to (Qin et al., 2024), we classify self-scoring techniques into data quality-based and data diversity-based methods. Data quality-based methods prioritize algorithms and metrics to score data items, selecting based on these scores, while data diversity-based methods prioritize dataset diversity. To examine the impact of self-scoring methods on LLMs' performance with vast IT data, we test recent methods on two benchmarks with millions of cases. Our experiments highlight three key points:

- Most self-scoring data selection methods perform similarly to random selection on large datasets. Although they show improvement on smaller datasets, their effectiveness decreases with larger and complex data. Some methods perform slightly better than random with certain LLMs, but balancing effectiveness and efficiency, random selection remains the best choice for large data sources.

- Data diversity holds more significance than data quality during the SFT phase. Data quality-based selection methods are more effective than data diversity-based methods when dealing with a small-scale dataset from a single source. However, when tackling multi-source data, only considering data quality is far from enough.

- Analyzing two IT datasets, we find using token length for data filtering ensures stable and efficient SFT results with large-scale IT data. Prior research (Liu et al., 2023) shows benefits of long text training for subjective evaluations like MTbench (Zheng et al., 2023) and AlpacaEval (Li et al., 2023c); we confirm its positive effect on objective tasks like Big-Bench-Hard (Suzgun et al., 2022). Although not always optimal for every language model, token length is beneficial in training long texts, notably for a weaker BASE model such as Llama3-8B.

2 Related Work

External-scoring Method. (Lu et al., 2023) proposed INSTAG, an open-set instruction tagging method using ChatGPT to generate tags for measuring instruction diversity/complexity in SFT. ALPAGASUS (Chen et al., 2023) model used ChatGPT to score instructions for threshold-based data selection. (Du et al., 2023) introduced a model-oriented selection approach considering instruction quality, coverage, and LLM capability. (Liu et al., 2023) developed DEITA, which iteratively enhanced data complexity or quality via ChatGPT and requested its evaluation. These models outperformed full-dataset baselines but rely heavily on external LLMs for scoring.

Self-scoring Method. (Li et al., 2023b) proposed an LLM self-directed method using IFD metrics to identify instruction pairs. DiverseEvol (Wu et al., 2023) enables models to independently select diverse subsets without external oversight. (Xia et al., 2024) introduced LESS, using gradient datastores to select instruction-tuning data. (Yin et al., 2024) proposed ZIP to favor low-compression-ratio subsets, while SelectIT (Liu et al., 2024) uses LLM uncertainty for efficient selection. Nuggets (Li et al., 2023d) employs perplexity-based one-shot scoring for high-quality data selection.

Specifically, in early research, data distillation had a similar goal to current SFT data selection, which was to filter out a small number of repre-

sentative data from large datasets (Lei and Tao, 2023). With the continuous development of LLMs, DEFT tasks in the past two years have focused on data distillation for specific tasks. This includes filtering data using feedback preferences during the reinforcement learning stage (Zhu et al., 2024) or selecting fine-tuning data for a specific task such as text editing (Das and Khetan, 2023). In contrast, SFT does not focus on specific tasks. Instead, it emphasizes unlocking various capabilities of LLMs through fine-tuning, such as code generation and logical reasoning. Therefore, the two approaches have slightly different emphases during the data selection stage.

3 Self-scoring strategies

In this paper, we focus on self-scoring methods that do not rely on external advanced LLMs to score data. We refer (Qin et al., 2024)'s work and categorize existing resourceful data selection methods into two main perspectives: data quality-based methods and data diversity-based methods.

3.1 Quality-based Selections

In this section, we introduce 4 methods based on data quality assessment and selection. "Quality" here refers primarily to the complexity, completeness, score, and influence of the datapoints. Different from (Qin et al., 2024), we believe that the influence of a datapoint in the target dataset is also a reflection of data quality, especially in practical scenarios, where we are required to deal with diverse tasks rather than a single task. We thus regard the influence as a quality category as well.

LESS (Xia et al., 2024) employed low-rank gradient similarity search for selecting influential data in target applications. Initially, a model was pretrained with LoRA (Hu et al., 2021) using a small subset $\mathcal{D}_{\text{warmup}} \subset \mathcal{D}$, after which the Adam LoRA gradient features were calculated and saved in a database. Then, a datastore of reduced-dimensional gradient features was established for reuse with various target tasks. For training points $\boldsymbol{x}$, they computed a d-dimensional projection of the LoRA gradient $\tilde{\nabla}\ell(\boldsymbol{x};\boldsymbol{\theta}_i) = \Pi^\top \hat{\nabla}\ell(\boldsymbol{x};\boldsymbol{\theta}_i)$, where $\Pi^\top$ uses a memory-efficient online implementation of random projections from (Park et al., 2023). For validation points $\boldsymbol{x}'$, $\tilde{\Gamma}(\boldsymbol{x}',\cdot) = \Pi^\top \hat{\Gamma}(\boldsymbol{x}',\cdot)$ was calculated, representing gradient values for $\boldsymbol{x}'$ across different optimization states. Finally, LESS evaluated $\max_j \text{Inf}_{\text{Adam}}(\boldsymbol{x}, \mathcal{D}_{\text{val}}^{(j)})$ over all validation subsets $\mathcal{D}_{val}$, choosing the top-scoring examples for $\mathcal{D}_{train}$.,

$$\text{Inf}_{\text{Adam}}(\boldsymbol{x}, \mathcal{D}_{\text{val}}^{(j)}) = \sum_{i=1}^{N} \bar{\eta}_i \frac{\langle \bar{\nabla}\ell(\mathcal{D}_{\text{val}}^{(j)};\boldsymbol{\theta}_i), \tilde{\Gamma}(\boldsymbol{x},\boldsymbol{\theta}_i)\rangle}{\|\bar{\nabla}\ell(\mathcal{D}_{\text{val}}^{(j)};\boldsymbol{\theta}_i)\| \|\tilde{\Gamma}(\boldsymbol{x},\boldsymbol{\theta}_i)\|} \quad (1)$$

IFD introduced the Instruction-Following Difficulty (IFD) score, a metric devised to evaluate the challenge each instructional sample presents (Li et al., 2023b). Given a (Q, A) pair, they calculated the ratio between $s(A)$ and $s(A|Q)$:

$$\text{IFD}(Q, A) = \frac{s(A|Q)}{s(A)} = \frac{-\frac{1}{N}\sum_{i=1}^{N} \log P(x_i^A | Q, x_1^A, x_2^A, ..., x_{i-1}^A)}{-\frac{1}{N}\sum_{i=1}^{N} \log P(x_i^A | x_1^A, ..., x_{i-1}^A)} \quad (2)$$

where $s(A)$ means Direct Answer Score, which measures LLM's ability to generate the answer alone. $s(A|Q)$ means Conditioned Answer Score, which is calculated by continuously predicting the next tokens given the instruction Q and their proceeding words.

The authors initially created 100 clusters from instruction embeddings and selected 10 instances per cluster according to the IFD score on a pretrained base LLM. They then trained this LLM for 1 epoch with these chosen datapoints. Post-training, they recalculated the IFD score of each datapoint in the entire training set $\mathcal{D}$ and ultimately chose the data with the highest IFD score as $\mathcal{D}_{train}$.

SelectIT identified high-quality IT data by analyzing the inherent uncertainty indicated by LLMs (Liu et al., 2024). It evaluated samples at three granular levels: token, sentence, and model level reflections. At the token level, SelectIT determined the probability of the following token (from 1 to K) using the rating prompt RP and the query-response pair E. The token with the highest probability was deemed the sample's quality measure. A higher $P'_{E^{base}}$ indicated greater LLM confidence.

$$E^{base} = \text{ arg max } P'_k, P'_k = \left(\frac{e^{P_k}}{\sum_{j=1}^{K} e^{P_j}} \right) \quad (3)$$

Here, P_k and P'_k denote the probability and softmax probability of token k, respectively. K represents the number of scores considered. In that study, the score tokens spanned from 1 to 5. To improve the reliability of quality assessment, SelectIT evaluated the average difference between the predicted token E^{base} and others, with larger differences indicating higher LLM confidence.

$$E^{token} = E^{base} \times \frac{1}{K-1} \sum_{i=1}^{K} |P'_i - P'_{E^{base}}| \quad (4)$$

At the sentence level, different prompts can notably influence LLM outcomes, so K semantically similar rating prompts $\{RP_0, RP_1, \ldots, RP_K\}$ were crafted, resulting in a set of quality scores $\{E_0^{token}, E_1^{token}, \ldots, E_K^{token}\}$.

$$E^{sent} = \frac{\mathbf{Avg}\{E_i^{token}\}_{i=1}^{K}}{1 + \alpha \times \mathbf{Std}\{E_i^{token}\}_{i=1}^{K}} \quad (5)$$

where $\mathbf{Avg}\{\cdot\}$ and $\mathbf{Std}\{\cdot\}$ denote the mean and standard deviation of E_i^{token}, respectively. K means the number of rating prompts RP.

For model level, SelectIT used N foundation models with parameter counts $\{\beta_1, \beta_2, \ldots, \beta_N\}$ and their respective sentence-level scores for a sample E being $\{E_0^{sent}, E_1^{sent}, \ldots, E_N^{sent}\}$, then the model-level score E_{model} was computed as follows.

$$E^{model} = \sum_{i=1}^{N} \left(\frac{\beta_i}{\sum_{j=1}^{N} \beta_j} \times E_i^{sent} \right) \quad (6)$$

where N means the number of the foundation models. It used E_{model} as the final evaluation of sample E in SelectIT.

Cross-entropy: Language models can be considered a form of compression, with LLMs showing strong capabilities in data compression empirically (Delétang et al., 2024). Compression efficiency is a stable and reliable assessment that is linearly related to the model's capabilities. It reflects the model's ability to extract relevant information and eliminate unnecessary elements, providing insight into the intrinsic capability of the language model (Huang et al., 2024; Wei et al., 2024).

The cross-entropy loss employed in the training of LLMs establishes a coherent relationship between LLMs and information compression of each query-response pair E.

$$\mathbb{E}_{x^E \sim \rho}[-\sum_{i=1}^{n} \log_2 \rho_{model}(x_i^E | x_{1:i-1}^E)] \quad (7)$$

Inspired by this foundational insight, we select data based on the cross-entropy of each datapoint, where the higher value of cross-entropy means the better quality.

3.2 Diversity-based Selections

In this section, we introduce methods that emphasize the diversity of instruction datasets, where diversity refers to the overall diversity of the entire training dataset.

DiverseEvol selectively sampled training subsets to enhance its performance iteratively (Wu et al., 2023). It identified distinct new data points in its current embedding space each iteration. For a dataset $\mathcal{D}$, DiverseEvol initially picked a random data pool P_0 and trained an initial model M_0. Each iteration involved: 1. Adding new data points $\mathcal{D}_t$ to P_{t+1} based on model M_t. 2. Training the next model M_{t+1} with updated P_{t+1}. The K-Center-Sampling method was used to choose k data points from candidates, maximizing their distance from existing training data.

$$\arg\max_{i \in X_t} \min_{j \in P_t} \Delta\left(\boldsymbol{x_i}, \boldsymbol{p_j}\right) \quad (8)$$

At each step, the input parameters to K-Center-Sampling were the model M_t, the current training pool P_t, and $\mathcal{D}_t$. The selection function K-Center-Sampling then outputs the new data point X_t, which was added to the training pool for the next iteration P_{t+1}.

The method **ZIP** identifies a negative correlation between model performance and the compression ratio of training data, often leading to reduced training loss. (Yin et al., 2024) introduced ZIP, a highly efficient and universal data selection approach for training LLMs, focusing on data subsets with low compression ratios.

It begins by determining the sample-level compression ratio for the dataset $\mathcal{D}$, with $\pi_{\mathcal{D}}$ representing data redundancy. In each cycle, it picks K_1 samples with the smallest $\pi_{\mathcal{D}_1}$ to create an initial pool $\mathcal{D}_{K_1}$. It then calculates the compression ratio of the combined set when adding each sample in $\mathcal{D}_{K_1}$ to the selected set $\mathcal{D}_{train}$, updating the redundancy $\pi_{\mathcal{D}_1}$. Based on sample scores in $\mathcal{D}_{K_1}$, ZIP chooses $\mathcal{D}_{K_2}$ samples with the smallest scores. Next, an empty set $\mathcal{D}_{K_3}$ is initialized, and the compression ratio for the union of $\mathcal{D}_{K_3}$ and each $\mathcal{D}_{K_2}$ sample is computed. The sample with the lowest ratio is added to $\mathcal{D}_{K_3}$ and removed from $\mathcal{D}_{K_2}$. Finally, each $\mathcal{D}_{K_3}$ sample is included in $\mathcal{D}_{train}$. The compression ratio $g(\mathcal{C}(D))$ in ZIP is computed as:

$$g(\mathcal{C}(D)) = \frac{\text{Bits}(D)}{\text{Bits}(\mathcal{C}(D))} \quad (9)$$

4 Experiment

4.1 Datasets

In practice, researchers often deal with large and imperfect datasets from diverse sources in SFT. This study, instead of using the usual IT datasets

like alpaca (Taori et al., 2023), uses two large-scale IT datasets at the million level, Openhermes2.5 (Teknium, 2023) and WildChat-1M (Zhao et al., 2024), to evaluate how current data selection methods perform with large datasets and to assess their performance in real-world scenarios.

Openhermes2.5 is introduced in (Teknium, 2023) with over 1 million entries, characterized by its extensive coverage and quality. It mainly includes generated guides and conversations from 16 sources, such as metamath (Yu et al., 2023), CamelAI (Li et al., 2023a), etc., covering topics like mathematics, programming and etc..

WildChat-1M from (Zhao et al., 2024) contains exclusively non-toxic user inputs and ChatGPT exchanges, totaling 1 million dialogues. About 25.53% involve GPT-4, the rest GPT-3.5, featuring varied interactions like ambiguous queries and political talks. This study extracts over 440k English dialogues from WildChat.

4.2 Benchmarks

To evaluate LLM capabilities, we explore various methods across downstream tasks. We use two datasets, GSM (Cobbe et al., 2021) and BBH (Suzgun et al., 2022), to test reasoning in the CoT setting (Wei et al., 2022). For code generation, we employ the HumanEval dataset (Chen et al., 2021) and reported pass@1 results. We gauge factual knowledge using MMLU (Hendrycks et al., 2021) with 5-shot results and assess instruction-following using IFEval (Zhou et al., 2023b) with strict and loose scores. Additionally, we use Open-Instruct scripts covering key benchmarks (Wang et al., 2023a; Ivison et al., 2023, 2024).

4.3 Implementation Details

Specifically, we leverage the widely-used LLaMA3-8B (AI@Meta, 2024) and Qwen2-7B (qwe, 2024) as our base models, and fine-tune them using the Llama-Factory framework (Zheng et al., 2024). We train these models for 3 epochs with a batch size of 128. Our training process employs a cosine learning rate scheduler beginning at $7e-6$, which decays to 0.1, warms to 0.01, and utilizes an input length of 4096. To replicate our baseline methods on Openhermes and WildChat, we adjust some original parameters and implementations to fit the large-scale datasets. The specific details of model reproduction are in Appendix A.1.

5 Discussion

5.1 Baseline Methods vs Random

This section replicates baseline methods for LLaMA3-8B and Qwen2-7B experiments on OpenHermes2.5, with results in Table 2 and WildChat results in Table 3. We evaluate these models with and without full dataset fine-tuning, using SFT data selection methods to pick 10,000 samples as per Section 4.3. We conduct 5 random runs and the outcomes are in the tables. Additionally, 50,000 samples from various methods are in Appendix Table 6, 7.

	Llama3-8B		Qwen2-7B	
	OpenHermes	WildChat	OpenHermes	WildChat
LESS	0.77	0.45	0.80	0.86
IFD	0.85	0.53	0.85	0.68
SelectIT	0.71	0.79	0.60	0.58
Entropy	0.92	0.46	0.78	0.30
Diverse	0.39	0.58	0.37	0.45
zip	0.55	0.36	0.42	0.31

Table 1: The P-values of the significance tests for each method against the results of five rounds of random selection.

As shown in Tables 2 and 3, no data selection methods significantly surpasses random sampling for large, varied IT datasets. Typically, baseline results fall within the range of five random runs, and some are even lower than the worst random outcome. For example, for Cross-Entropy on Qwen2-7B with Openhermes2.5, the average result is 54.02, which is notably less than the lowest random score of 57.04. We also applied the Mann-Whitney U test, using a right-tailed hypothesis that baseline scores exceed random ones, and documented the p-values in Table 1. All methods had p-values over 0.05, indicating no baseline method outperformed random selection.

Based on the experimental results, **when dealing with an extensive SFT dataset, it is more efficient to randomly select training data instead of spending significant time and resources to meticulously choose seemingly optimal training data.** Random selection reduces costs and yields superior training results.

5.2 Quality vs Diversity

Tables 2 and 3 demonstrate that the diversity-based selection strategies outperforms the quality-based one. To examine whether prioritizing diversity over data quality improves data selection, we designed a supplementary experiment by incorporat-

	Qwen2-7B							Llama3-8B						
	BBH	GSM	CODE	MMLU	IFEVAL		AVG	BBH	GSM	CODE	MMLU	IFEVAL		AVG
	3 shot	8 shot	pass 1	5 shot	strict	loose		3 shot	8 shot	pass 1	5 shot	strict	loose	
Base	59.07	72.40	55.67	70.20	28.84	31.24	52.90	60.93	55.12	37.59	65.30	19.41	21.07	43.24
all data	61.39	80.12	63.32	68.50	40.85	44.18	59.73	63.33	73.24	46.43	63.90	46.40	49.72	57.17
Random 1	59.72	82.41	62.10	68.30	33.27	36.41	57.04	64.72	53.90	45.21	63.20	39.19	43.62	51.64
Random 2	61.48	**83.47**	64.33	67.90	38.08	40.30	59.26	60.83	56.86	**48.99**	62.70	41.77	45.47	52.77
Random 3	**61.85**	81.65	62.90	68.10	36.78	38.45	58.29	63.43	59.74	46.83	62.70	43.25	46.21	53.69
Random 4	61.20	82.71	59.27	68.00	36.60	39.19	57.83	63.98	59.59	45.18	63.80	44.36	47.13	**54.01**
Random 5	61.30	82.71	62.23	**68.90**	35.86	37.71	58.12	62.31	56.10	42.07	63.50	**44.55**	**48.80**	52.89
LESS	61.20	81.65	53.26	67.60	32.16	37.15	55.50	61.39	57.70	41.43	**64.20**	38.08	41.96	50.79
IFD	57.96	79.23	**68.48**	56.70	33.27	35.12	55.13	57.41	53.53	32.41	59.90	43.07	45.84	48.69
SelectIT	59.17	80.44	66.46	67.20	35.86	38.82	57.99	62.59	**61.56**	42.38	63.60	38.45	42.14	51.79
Entropy	61.30	55.04	61.04	**68.90**	37.34	40.48	54.02	58.61	50.72	44.02	61.40	32.90	37.89	47.59
Diverse	61.11	81.73	61.71	68.65	**40.85**	**43.44**	**59.58**	**65.00**	56.25	44.51	63.84	43.99	47.13	53.45
ZIP	60.65	80.52	66.10	68.60	37.15	39.56	58.76	63.98	59.67	40.70	62.60	43.81	46.58	52.89

Table 2: The overall results (%) on a variety of downstream tasks based on Openhermes2.5 dataset. CODE means HumanEval, Random n denotes the nth random selection. Except for fine-tuning with the entire Openhermes dataset, the bold numbers indicate the best score of each part, and the underlined numbers indicate the second highest score.

ing a K-means clustering process on the OpenHermes dataset. Instead of selecting data based solely on method scores, we choose higher-scoring data within each cluster to boost the final training set's diversity.

Table 5 illustrates that integrating the K-means clustering with quality-based selection methods enhances the effectiveness for most approaches. Notably, Cross Entropy on both Llama3 and Qwen2 models shows improvement over 5% and 3%, respectively, when K-means is used to diversify the data. This suggests that for a large-scale IT dataset, **data diversity holds more importance than data quality**. This also clarifies why random selection often outperforms most SFT data selection methods, as the random process preserves the dataset's original distribution and diversity to the greatest possible extent.

5.3 Baseline Analysis

In this part, we mainly analyze several methods and try to find the reasons why these methods fail in large-scale data sets and why these methods are not applicable to practical applications.

The lack of availability of **Less** is primarily evident in how its influence score is calculated. Since it requires computing the score for the final data point in the target task, it is essential to meticulously design a target set for each task to filter the data. However, in practical applications, we face a variety of training tasks that require our target data to be comprehensive and diverse. Hence, the effectiveness of LESS is strongly related to the quality of $\mathcal{D}_{val}$.

The **IFD** approach determines the ultimate IFD score by evaluating the perplexity (ppl) of the response. However, the length of the data significantly affects the ppl value. In particular, shorter data tend to produce excessively high ppl values, which contradicts with our expected results. Ultimately, we note that the IT data instructions selected by the IFD approach are quite brief, averaging merely 42 tokens on Openhermes, which aligns with the findings reported by (Liu et al., 2023).

SelectIT can perform well at the model level, but it necessitates combining LLMs with various sizes to score the data. As IT datasets become larger, the computational cost required for LLMs with more parameters tends to increase exponentially, which limits their applicability to extensive datasets.

Cross-entropy is influenced by the length of responses. Typically, cross-entropy favors data with lengthy responses, whereas it shows no specific preference towards instructions. Consequently, the training samples will include simple instructions but extensive responses.

We exclude **NUGGETS** (Li et al., 2023d) as a baseline due to its extensive computational de-

	Qwen2-7B							Llama3-8B						
	BBH 3 shot	GSM 8 shot	CODE pass 1	MMLU 5 shot	IFEVAL strict	IFEVAL loose	AVG	BBH 3 shot	GSM 8 shot	CODE pass 1	MMLU 5 shot	IFEVAL strict	IFEVAL loose	AVG
Base	59.07	72.40	55.67	70.20	28.84	31.24	52.90	60.93	55.12	37.59	65.30	19.41	21.07	43.24
all data	62.87	80.82	62.84	68.70	45.84	48.80	61.65	63.70	56.94	47.44	63.30	46.40	49.72	54.58
Random 1	61.30	**82.64**	61.98	68.10	40.30	42.33	59.44	63.70	56.48	51.92	63.30	39.37	41.95	52.79
Random 2	60.93	81.96	61.43	67.50	38.63	40.67	58.52	62.41	52.62	**49.33**	64.00	44.18	46.77	53.22
Random 3	60.28	**82.64**	62.07	68.30	41.04	42.88	59.54	63.52	58.38	43.90	64.10	42.33	45.29	52.92
Random 4	61.11	80.36	65.46	67.50	37.34	40.67	58.74	63.33	55.42	51.10	64.50	41.96	44.55	53.48
Random 5	61.57	81.50	60.27	68.20	41.77	43.99	59.55	**64.91**	60.27	48.66	64.30	42.14	45.84	54.35
LESS	52.59	60.50	61.19	68.00	38.82	41.77	53.81	63.43	57.01	50.43	64.50	40.85	44.92	53.52
IFD	60.56	76.27	65.24	68.00	36.23	38.26	57.43	63.33	59.29	47.16	**64.60**	40.30	43.81	53.08
SelectIT	60.37	82.34	64.97	**68.50**	36.97	39.19	58.72	61.48	53.22	46.01	63.20	40.11	42.88	51.15
Entropy	60.37	81.96	62.90	68.40	**42.51**	**46.21**	60.39	63.15	56.10	47.71	63.00	45.10	49.54	54.10
Diverse	61.02	80.82	65.09	67.33	41.04	42.88	59.70	62.59	53.30	33.48	64.46	**47.87**	**50.65**	52.06
ZIP	**62.59**	81.80	**68.17**	68.00	40.11	42.33	**60.50**	62.31	**60.96**	46.58	64.50	45.10	48.06	**54.59**

Table 3: The overall results (%) on a variety of downstream tasks based on WildChat dataset. CODE means HumanEval, Random n denotes the nth random selection. Except for fine-tuning with the entire Openhermes dataset, the bold numbers indicate the best score of each part, and the underlined numbers indicate the second highest score.

	Qwen2-7B							Llama3-8B						
	BBH 3 shot	GSM 8 shot	CODE pass 1	MMLU 5 shot	IFEVAL strict	IFEVAL loose	AVG	BBH 3 shot	GSM 8 shot	CODE pass 1	MMLU 5 shot	IFEVAL strict	IFEVAL loose	AVG
OpenHermes	60.65	80.74	60.18	68.33	37.89	41.40	58.20	64.63	61.33	45.70	64.41	48.43	52.87	56.23
WildChat	61.67	81.05	59.21	67.82	39.56	42.14	58.58	66.11	60.35	51.16	63.91	43.81	47.69	55.51

Table 4: The overall results (%) of token length selection.

mands, requiring over 2,000 hours on 40 A100 80G GPUs. Given this high time cost, we decide to abandon this method.

The diversity-based approach usually outperforms the quality-based selection methods, however, one main issue with the diversity-based approach is its time and memory consumption.

To replicate **DiverseEvol**, we used 8 A100 80G GPUs across 3 iterations, each lasting 1-2 days, totaling 5-7 days to select the final subset. When dealing with large-scale data sets, the results often fall within the random range, though optimal results occur sporadically. This may be due to modifications in our implementation to address memory constraints during replication (see Section 4.3), which may have slightly diminished the method's performance. In contrast, **ZIP** does not need GPU resources, but the computing process is greedy. It incrementally adds 100 data at a time to the final training subset. For large data scales, it takes approximately 7 days to select 50,000 data. In addition, ZIP serves as a data selection method that operates independently of the model, meaning that the selected data cannot be adaptively tuned on the basis of the model. As illustrated in Tables 2 and 3, the data chosen by ZIP in OpenHermes perform poorly in both Llama3-8B and Qwen2-7B, whereas the data selected in WildChat exhibit the best performance across these models.

Moreover, we attempt to utilize **DQ** (Zhou et al., 2023a) as our baseline method. However, DQ uses a submodular strategy to choose a subset by optimizing submodular gains within the feature space. When dealing with millions of data points, it requires more than 1TB memory resources. Eventually, we decide to forgo this approach.

5.4 Which method is the best?

By examining the average results, we notice that the majority of methods perform better with WildChat as the data source compared to OpenHermes, as illustrated in Figure 2, which is rather unex-

	Qwen2-7B							Llama3-8B						
	BBH 3 shot	GSM 8 shot	CODE pass 1	MMLU 5 shot	IFEVAL strict	IFEVAL loose	AVG	BBH 3 shot	GSM 8 shot	CODE pass 1	MMLU 5 shot	IFEVAL strict	IFEVAL loose	AVG
LESS	61.20	81.65	53.26	67.60	32.16	37.15	55.50	61.39	57.70	41.43	64.20	38.08	41.96	50.79
IFD	57.96	79.23	68.48	56.70	33.27	35.12	55.13	57.41	53.53	32.41	59.90	43.07	45.84	48.69
SelectIT	59.17	80.44	66.46	67.20	35.86	38.82	57.99	62.59	61.56	42.38	63.60	38.45	42.14	51.79
Entropy	61.30	55.04	61.04	68.90	37.34	40.48	54.02	58.61	50.72	44.02	61.40	32.90	37.89	47.59
LESS$_{km}$	61.30	81.96	54.63	67.79	34.38	38.26	**56.39**	60.93	50.27	48.11	63.97	39.74	44.55	**51.26**
IFD$_{km}$	60.19	78.77	59.70	66.81	30.31	31.79	54.60	60.74	58.98	40.37	62.95	40.67	42.70	**51.07**
SelectIT$_{km}$	60.93	82.34	61.04	67.85	36.78	39.19	**58.02**	62.96	59.36	40.85	63.43	39.74	43.07	51.57
Entropy$_{km}$	60.37	81.12	59.27	68.55	35.67	38.45	**57.24**	61.02	61.64	48.32	61.12	39.00	43.99	**52.52**

Table 5: Overall results (%) for various downstream tasks are based on the Openhermes2.5 dataset. The notation Method$_{km}$ refers to the method incorporating the k-means process. Bold numbers represent the average performance gain following the addition of the K-means phase.

Figure 2: The average score (%) of each methods on Llama3 and Qwen2.

pected. Nonetheless, from a quality perspective, WildChat's conversation data tends to be noisy, particularly since the context of multiple conversation rounds is sometimes unrelated, while OpenHermes's data quality should be substantially higher than WildChat. However, the performance patterns for these data sources are contrary to our predictions. WildChat's average token length is 1142, compared to 354 for OpenHermes. Inspired by (Shen, 2024), we designed a new experiment focused on selecting data by token length. We applied K-Means to form N clusters, then chose a data quantity from each cluster proportional to its size, based on token length. Results are in Table 4.

Based on Table 4, it is evident that using token length as the criterion for data selection generally yields optimal results. Specifically, for Llama3, regardless of whether the data source is OpenHermes or WildChat, the results are superior to those achieved by other methods. In addition, the average score on WildChat (55.51) surpasses that obtained by fine-tuning with the entire dataset (54.58). Since random selection may not ensure the best fine-tuning results, we believe that **selecting data by token length can stably obtain a relatively high training benefit, reduce the uncertainty caused by randomness, and reduce costs.** This approach is particularly beneficial for BASE language models which generally have limited capabilities, as they tend to derive the most significant benefits from training on longer texts. Notably, both Qwen2 (qwe, 2024) and Llama3 (AI@Meta, 2024) incorporate long-text training components in their pre-training stages. Based on this observation, we posit that with the continuous iteration of foundational models, the advantages of length-based data selection will gradually diminish.

6 Conclusion

In this study, we rethinking whether SFT data selection methods can work when they are required to handle large-scale IT datasets. We replicate existing self-scoring data selection methods on million-scale datasets and observe that most hardly outperform random selection. Additionally, during the SFT phase, data diversity matters more than quality. Token length proves a better quality metric for SFT data selection than other detailed metrics.

7 Limitations

Due to financial limitations, the External-scoring Method was not implemented as a comparative approach in this study. We were unable to identify a data selection technique that universally applies to all LLMs. While accounting for both temporal costs and model effectiveness, it appears that token length typically yields optimal outcomes; however, this method is not suitable for every model.

8 Ethics Statement

The primary aim of this study is to select specific portions of data from existing open-source public datasets to be used in the supervised fine-tuning of LLMs. We have chosen two datasets for this purpose: OpenHermes2.5 and WildChat. OpenHermes2.5 comprises various general open-source datasets that are free from security or ethical concerns. Meanwhile, the WildChat dataset has been curated to exclude toxic user inputs, thus guaranteeing its safety.

References

2024. Qwen2 technical report.

AI@Meta. 2024. Llama 3 model card.

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

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Devleena Das and Vivek Khetan. 2023. Deft: Data efficient fine-tuning for pre-trained language models via unsupervised core-set selection. *arXiv preprint arXiv:2310.16776*.

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. 2024. Language modeling is compression. In *ICLR*.

Qianlong Du, Chengqing Zong, and Jiajun Zhang. 2023. Mods: Model-oriented data selection for instruction tuning. *arXiv preprint arXiv:2311.15653*.

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

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

Yuzhen Huang, Jinghan Zhang, Zifei Shan, and Junxian He. 2024. Compression represents intelligence linearly. *Preprint*, arXiv:2404.09937.

Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. 2024. Unpacking dpo and ppo: Disentangling best practices for learning from preference feedback. *Preprint*, arXiv:2406.09279.

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

Shiye Lei and Dacheng Tao. 2023. A comprehensive survey of dataset distillation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):17–32.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. Camel: Communicative agents for "mind" exploration of large scale language model society. *Preprint*, arXiv:2303.17760.

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

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

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

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

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

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

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

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. 2023. Trak: Attributing model behavior at scale. In *International Conference on Machine Learning (ICML)*.

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

Yulei Qin, Yuncheng Yang, Pengcheng Guo, Gang Li, Hang Shao, Yuchen Shi, Zihan Xu, Yun Gu, Ke Li, and Xing Sun. 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*.

Ming Shen. 2024. Rethinking data selection for supervised fine-tuning. *arXiv preprint arXiv:2402.06094*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

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

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

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

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

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023b. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Lai Wei, Zhiquan Tan, Chenghai Li, Jindong Wang, and Weiran Huang. 2024. Large language model evaluation via matrix entropy. *arXiv preprint arXiv:2401.17139*.

Shengguang Wu, Keming Lu, Benfeng Xu, Junyang Lin, Qi Su, and Chang Zhou. 2023. Self-evolved diverse data sampling for efficient instruction tuning. *arXiv preprint arXiv:2311.08182*.

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

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei

Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Mingjia Yin, Chuhan Wu, Yufei Wang, Hao Wang, Wei Guo, Yasheng Wang, Yong Liu, Ruiming Tang, Defu Lian, and Enhong Chen. 2024. Entropy law: The story behind data compression and llm performance. *arXiv preprint arXiv:2407.06645*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

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

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

Daquan Zhou, Kai Wang, Jianyang Gu, Xiangyu Peng, Dongze Lian, Yifan Zhang, Yang You, and Jiashi Feng. 2023a. Dataset quantization. *arXiv preprint arXiv:2308.10524*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023b. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

Kun Zhou, Beichen Zhang, Jiapeng Wang, Zhipeng Chen, Wayne Xin Zhao, Jing Sha, Zhichao Sheng, Shijin Wang, and Ji-Rong Wen. 2024b. Jiuzhang3.0: Efficiently improving mathematical reasoning by training small data synthesis models. *arXiv preprint arXiv:2405.14365*.

Liang Zhu, Feiteng Fang, Yuelin Bai, Longze Chen, Zhexiang Zhang, Minghuan Tan, and Min Yang. 2024. Deft: Distribution-guided efficient fine-tuning for human alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15318–15331.

A Appendix

A.1 Model Reproduction Details

In term of LESS, individual models are built and trained on specific tasks. However, in practical applications, our goal is to train a model that enhances performance across various scenarios. Thus, given that the two datasets we select are both extensive and diverse, we randomly select 1000 data points from each dataset as $\mathcal{D}_{val}$. Additionally, due to the volume of our data, we randomly pick 10,000 data points for warm-up training, differing from the method described in (Xia et al., 2024).

As for IFD, we initially generate 1000 clusters on instruction embeddings, which differs from the settings given in (Li et al., 2023b). For SelectIT, we adopt model-level selection as the final strategy for the Qwen2 model and evaluate the model-level score on Qwen2-1.5B and Qwen2-7B. While for Llama3, we employ sentence-level selection as the final approach. Considering that the Llama3 family only has two public variants, Llama3-8B and Llama3-70B, and to mitigate time costs, we compute the score based solely on Llama3-8B.

Within DiverseEvol, during each iteration's K-Center-Sampling stage, data points are selected based on maximizing their distance to the nearest existing training data points, one at a time, until the desired count is reached. Consequently, it is essential to maintain a $n \times n$ float-type matrix for the entire computation, where n represents the dataset size. Given that our OpenHermes dataset exceeds 1 million entries, the matrix calculation would require more than 1 terabyte of memory. Therefore, we revised this part to select all required data points once for each iteration, which significantly reduces the memory requirement.

A.2 Other Results

In this section, table 6, 7 includes training results of various methodologies with a training dataset comprising 50,000 entries 6, 7.

	Qwen2-7B							Llama3-8B						
	BBH	GSM	CODE	MMLU	IFEVAL		AVG	BBH	GSM	CODE	MMLU	IFEVAL		AVG
	3 shot	8 shot	pass 1	5 shot	strict	loose		3 shot	8 shot	pass 1	5 shot	strict	loose	
Base	59.07	72.40	55.67	70.20	28.84	31.24	52.90	60.93	55.12	37.59	65.30	19.41	21.07	43.24
all data	61.39	80.12	63.32	68.50	40.85	44.18	59.73	63.33	73.24	46.43	63.90	46.40	49.72	57.17
Random_1	**62.87**	80.67	62.44	68.33	34.75	38.08	57.86	63.89	64.37	46.19	62.75	45.10	49.72	55.34
Random_2	61.11	80.82	65.76	68.12	38.08	40.67	59.09	62.13	**66.57**	47.32	61.57	46.58	49.54	**55.62**
Random_3	61.02	81.35	60.15	68.54	38.63	40.85	58.42	**65.65**	63.53	44.05	61.96	42.51	46.21	53.99
Random_4	60.37	80.06	55.98	68.95	37.34	40.30	57.17	62.78	62.40	45.12	62.41	**47.87**	50.83	55.24
Random_5	60.19	80.14	63.29	**69.16**	38.08	40.85	58.62	64.72	65.13	45.18	62.51	45.47	49.17	55.36
LESS	60.46	80.29	58.66	67.40	39.00	43.25	58.18	61.02	57.85	17.01	63.01	40.30	46.40	47.60
IFD	57.50	80.52	67.13	66.79	35.86	38.08	57.65	61.94	52.84	44.63	63.36	41.04	43.99	51.30
SelectIT	60.56	79.98	62.77	67.96	36.04	39.00	57.72	61.20	64.22	40.03	62.40	41.96	44.92	52.46
Entropy	60.83	77.56	59.24	69.02	36.78	39.56	57.17	60.65	55.50	**49.02**	57.51	47.13	**51.02**	53.47
Diverse	61.67	81.35	61.89	68.60	**44.55**	**46.40**	**60.74**	63.33	61.11	48.75	**63.62**	46.21	49.17	55.37
zip	59.81	82.03	**68.48**	68.08	35.67	38.26	58.72	63.89	57.92	42.65	62.58	43.25	46.95	52.87
LESS_{km}	61.20	81.88	54.51	67.77	32.90	36.60	55.81	61.02	59.44	47.04	63.35	42.14	47.32	53.39
IFD_{km}	59.81	78.92	60.55	67.09	28.65	31.24	54.38	63.43	63.23	43.41	61.19	40.11	43.81	52.53
SelectIT_{km}	61.20	81.20	66.52	69.10	34.57	38.45	58.51	61.85	61.49	45.76	61.64	43.44	48.43	53.77
Entropy_{km}	61.02	80.82	66.04	68.25	36.78	39.37	58.71	61.85	64.22	48.66	61.85	42.70	46.58	54.31
Length_{km}	60.46	**83.62**	63.35	68.79	38.26	41.59	59.35	65.09	62.70	47.29	62.73	45.10	49.17	55.35

Table 6: The comprehensive results (%) on various downstream tasks using OpenHermes. Mention that CODE means Humaneval. Algorithm_{km} means the algorithm has a Kmeans process, and Random_x denotes the $_x$th random selection. The bold numbers indicate the best avg score of each part, and the underlined numbers indicate the second highest score.

	Qwen2-7B							Llama3-8B						
	BBH	GSM	CODE	MMLU	IFEVAL		AVG	BBH	GSM	CODE	MMLU	IFEVAL		AVG
	3 shot	8 shot	pass 1	5 shot	strict	loose		3 shot	8 shot	pass 1	5 shot	strict	loose	
Base	59.07	72.40	55.67	70.20	28.84	31.24	52.90	60.93	55.12	37.59	65.30	19.41	21.07	43.24
all data	62.87	80.82	62.84	68.70	45.84	48.80	61.65	63.70	56.94	47.44	63.30	46.40	49.72	54.58
Random_1	61.85	81.50	60.55	68.02	40.48	42.70	59.18	63.61	55.72	48.90	64.07	42.51	45.66	53.41
Random_2	60.74	82.03	58.72	68.05	40.67	44.36	59.10	61.76	54.66	50.95	63.38	42.88	46.03	53.28
Random_3	59.07	81.35	64.45	67.63	41.77	44.92	59.87	63.98	55.42	**53.11**	63.33	43.81	46.77	54.40
Random_4	**62.41**	82.34	60.95	68.43	42.51	45.10	60.29	63.70	58.91	50.09	63.84	43.62	46.03	54.37
Random_5	61.30	82.49	59.05	67.60	42.70	44.92	59.68	64.54	55.65	49.91	64.16	42.70	45.84	53.80
LESS	58.80	81.35	66.95	68.10	41.04	43.99	60.04	63.43	57.01	50.43	**64.50**	40.85	44.92	53.52
IFD	59.44	81.50	66.46	67.90	38.45	40.85	59.10	63.33	59.29	47.16	64.60	40.30	43.81	53.08
SelectIT	60.74	**84.23**	60.49	**69.24**	41.04	44.36	60.02	61.48	53.22	46.01	63.20	40.11	42.88	51.15
Entropy	61.02	81.96	60.88	68.40	43.07	46.58	60.32	61.48	55.34	48.90	64.02	47.50	**51.02**	54.71
Diverse	59.81	82.03	67.10	68.00	41.77	44.36	60.51	**65.09**	56.18	38.81	63.03	44.36	47.13	52.43
zip	59.91	79.83	71.04	67.97	42.88	45.84	**61.25**	64.72	57.16	41.49	61.54	**45.84**	48.43	53.20
LESS_{km}	59.54	80.89	67.84	68.20	**43.62**	**46.95**	61.17	61.94	54.74	48.99	64.10	43.99	46.95	53.45
IFD_{km}	59.26	80.67	**68.41**	68.13	41.77	43.99	60.37	62.69	56.10	48.63	63.02	40.85	42.70	52.33
SelectIT_{km}	60.46	83.17	59.39	68.79	39.93	43.07	59.14	61.20	54.89	45.88	63.50	43.99	48.06	52.92
Entropy_{km}	60.93	82.79	59.82	67.01	39.19	42.14	58.65	63.06	58.45	45.73	63.85	41.04	45.10	52.87
Length_{km}	61.30	79.76	59.76	68.19	42.88	45.29	59.53	62.41	**60.05**	49.82	64.23	45.47	48.80	**55.13**

Table 7: The comprehensive results (%) on various downstream tasks using WildChat. Mention that CODE means Humaneval. Algorithm_{km} means the algorithm has a Kmeans process, and Random_x denotes the $_x$th random selection. The bold numbers indicate the best avg score of each part, and the underlined numbers indicate the second highest score.