Boosting LLM via Learning from Data Iteratively and Selectively

Qi Jia [1] **Siyu Ren** [2] **Ziheng Qin** [1] **Fuzhao Xue** [3] **Jinjie Ni** [1] **Yang You** [1]

Abstract

Datasets nowadays are generally constructed from multiple sources and using different synthetic techniques, making data de-noising and de-duplication crucial before being used for post-training. In this work, we propose to perform instruction tuning by iterative data selection (ITERIT). We measure the quality of a sample from complexity and diversity simultaneously. Instead of calculating the complexity score once for all before fine-tuning, we highlight the importance of updating this model-specific score during fine-tuning to accurately accommodate the dynamic changes of the model. On the other hand, the diversity score is defined on top of the samples' responses under the consideration of their informativeness. ITERIT integrates the strengths of both worlds by iteratively updating the complexity score for the top-ranked samples and greedily selecting the ones with the highest complexity-diversity score. Experiments on multiple instruction-tuning data demonstrate consistent improvements of ITERIT over strong baselines. Moreover, our approach also generalizes well to domain-specific scenarios and different backbone models. All resources will be available at https://github.com/JiaQiSJTU/IterIT.

1. Introduction

Instruction tuning (Ouyang et al., 2022) is an important stage for large language models (LLMs) after the knowledge-centric pre-training. It tailors pre-trained LLMs from a massive knowledge reservoir to a useful knowledge provider by training on instruction-following data (Wang et al., 2022), endowing a significant boost on the LLMs' performance. Data synthesis techniques (Wang et al., 2023) play a key role in constructing such data. These techniques facilitate post-training in different domains due to their scalable nature, but meanwhile introduce more noises and duplications into the training data. Zhou et al. (2024) find that models fine-tuned with only around 1K manually curated high-quality samples already demonstrate a strong generalization ability in downstream tasks. This finding encourages research into identifying the valuable subset for instruction tuning, which can lead to competitive or even superior performance while significantly reducing training costs.

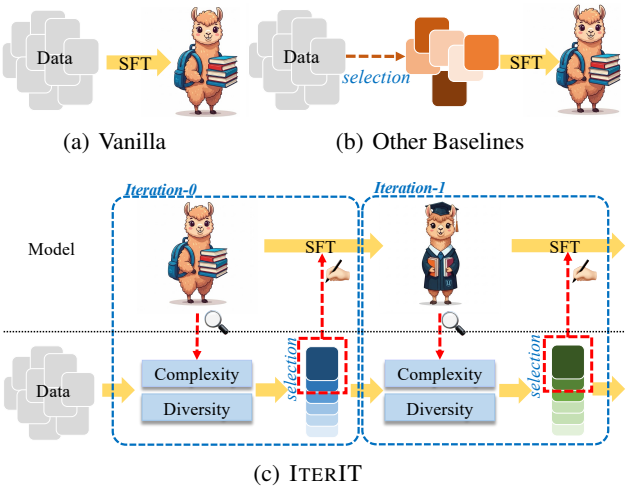


Figure 1. Illustrations of Vanilla, other baselines and ITERIT . Grey boxes represent the training data that hasn't been assessed, which will be ranked by different metrics, i.e., the colored boxes. The red arrows in ITERIT emphasize the collaboration between the model and the data. In other words, the model will supervise the data selection process, while the selected samples will be used to update the model's parameters.

To automate the data selection process, certain studies (Chen et al., 2023; Lu et al., 2023) employ powerful proprietary large language models, such as ChatGPT, to assess the quality of each sample based on predefined prompts. Nonetheless, these prompting methods incur additional costly expenses and lack interpretability. Liu et al. (2024b) developed scoring models using labels gathered from ChatGPT, and various approaches have been proposed to evaluate quality, either by leveraging a range of foundation models (Liu et al., 2024a) or by focusing on a single model itself (Li et al., 2024c;b;d). Meanwhile, some research (Shen, 2024; Zhao et al., 2024) also reveals that choosing samples by

[1]National University of Singapore [2]Meituan [3]Work done in National University of Singapore, now in Google DeepMind. Correspondence to: Yang You <youy@comp.nus.edu.sg>.

their response lengths is a simple but tough-to-beat baseline, even outperforming the elaborated metrics. With a deeper understanding of this task, researchers gradually reach a consensus on the importance of balancing the complexity and diversity of a selected subset (Lu et al., 2023; Liu et al., 2024b). Recent work by Wu et al. (2024a) has incorporated a complexity score that gauges the perplexity changes in the response, as derived from Li et al. (2024b), and a TF-IDF-based diversity score computed on the instructions, leading to further enhancements.

However, we recognize that previous approaches have not fully harnessed the potential of both complexity and diversity metrics, nor have they truly integrated the strengths of both. On one hand, data is only selected at the outset of the fine-tuning process, which precludes model-specific scores from capturing the dynamic changes in the model. Based on our preliminary analysis using the instruction-following difficulty (IFD) score (Li et al., 2024b), 55.31% of the samples in the top 5% of the selected subset after the first fine-tuning epoch were not included in the initial selection at the beginning of the fine-tuning process. On the other hand, previous studies (Li et al., 2024c; Wu et al., 2024a) suggest that the topic of a sample is predominantly influenced by its instruction. However, disparate instructions can result in similar and less informative responses. Since models are optimized using losses computed from response tokens, this limitation may impede the model's ability to function as an effective assistant.

To tackle the aforementioned shortcomings, we propose ITERIT, which boosts LLMs' instruction-following performance through iterative data selection. An illustration with comparisons is in Fig. 1. Specifically, we employ the IFD metric to assess the complexity of each sample in a coarse-to-fine manner. Once the complexity scores are computed for all samples, we retain only the top-ranked ones for further selection, filtering out the remainder. Additionally, we compute the sum of TF-IDF features on each sample's response to determine the diversity score for each sample, aiming to identify the most diverse subset of informative responses. Once a sample is chosen, the diversity scores of the remaining samples are updated with weight decay applied to the features that have already been covered. Based on these two criteria, before each fine-tuning epoch, ITERIT will update the complexity score of the reserved samples and greedily select a subset with the highest complexity-diversity score, continuing until the data of the specified budget is gathered. In extensive experiments conducted on instruction-tuning datasets of varying qualities, our ITERIT demonstrated remarkably better performance, achieving results superior to those of a model trained with the full dataset using only 5% of the data at each epoch, and consistently outperforming competitive baselines.

We summarize our contributions as follows:

- We introduce ITERIT, a new approach for instruction tuning by iterative data selection (Sec. 2). ITERIT shows favorable performance on tailoring a foundation language model to a useful assistant across different instruction-following datasets with extensive experiments (Sec. 4.1).
- Through experiments conducted on domain-specific datasets and other backbone LLMs, ITERIT has shown strong generalization capabilities and consistently delivered improvements over baselines (Sec. 4.2&4.3).
- Through ablation studies and analysis of the selected data, we demonstrate that ITERIT unleashes the power of complexity and diversity metrics, and underscores the critical importance of data-model collaboration during post-training (Sec. 5).

2. Approach

The goal of supervised fine-tuning (SFT) is to endow pre-trained foundation models with the ability to comprehend and follow user instructions. Let D denote the complete dataset containing N samples $\{X_i, Y_i\}|_{i=0}^{N-1}$, where $X_i = \{x_{i,0}, ..., x_{i,A-1}\}$ refers to the instruction and $Y_i = \{y_{i,0}, ..., y_{i,B-1}\}$ refers to the answer. A and B denote the corresponding number of tokens. The process of supervised fine-tuning is to maximize the model's likelihood of generating answer Y_i based on instruction X_i, i.e., $P_\theta(Y_i|X_i)$, where θ represents the LLM's parameters.

We propose ITERIT, the core of which lies in a novel iterative data selection algorithm, with enhancements on measuring samples from both the complexity and diversity aspects. Our approach relies solely on the model being trained itself with affordable computation for data selection. An illustration is shown in Fig. 1(c), with further details elaborated in the following sections.

2.1. Complexity Measurement

We characterize the complexity of a sample through instruction-following difficulty (IFD), as proposed by Li et al. (2024b). This metric quantifies the connection between X_i and Y_i by assessing the perplexity of Y_i using the model's prior knowledge directly, which is expressed as:

$$\mathrm{PPL}_{\mathrm{prior}}^{i,\theta} = \exp\left(-\frac{1}{B}\sum_{k=0}^{B-1}\log p_\theta(y_{i,k}|\boldsymbol{y}_{i,<k})\right), \quad (1)$$

and further conditioned on the corresponding instruction, as shown below:

$$\mathrm{PPL}_{\mathrm{cond}}^{i,\theta} = \exp\left(-\frac{1}{B}\sum_{k=0}^{B-1}\log p_\theta(y_{i,k}|\boldsymbol{y}_{i,<k}, X_i)\right). \quad (2)$$

The complexity score is then defined as the ratio of $\mathrm{PPL}_{\mathrm{cond}}^{i,\theta}$ to $\mathrm{PPL}_{\mathrm{prior}}^{i,\theta}$:

$$S_{\mathrm{COM}}^{i,\theta} = \frac{\mathrm{PPL}_{\mathrm{cond}}^{i,\theta}}{\mathrm{PPL}_{\mathrm{prior}}^{i,\theta}}. \quad (3)$$

This score serves to evaluate the effectiveness of instruction X_i in facilitating the generation of Y_i. A lower value of $S_{\mathrm{COM}}^{i,\theta}$ suggests that the model can successfully follow the user's instruction without the need for additional training on that specific sample, while a higher value signifies greater complexity. Ideally, the following condition should be held:

$$\forall i, \theta,\ \mathrm{PPL}_{\mathrm{cond}}^{i,\theta} < \mathrm{PPL}_{\mathrm{prior}}^{i,\theta}, \quad (4)$$

The rationale behind this is that the entropy of Y_i should decrease when provided with more relevant information. If this condition is not met, the instruction-response pair is considered to be unaligned for the model.

2.2. Diversity Measurement

We base the diversity score of a selected subset on the informativeness of individual samples and the degree of information overlap among these samples. Specifically, the informativeness of a sample is predominantly manifested through its response, which conveys not only the topics of the sample but also the attitude in the response. A response rich in detailed information is generally more valuable and preferred by humans. Drawing inspiration from Wu et al. (2024a), we quantify the diversity score using the TF-IDF score of n-grams present in the response Y_i. Mathematically, the TF-IDF for a single n-gram g is defined as:

$$\begin{aligned}\text{TF-IDF}(g, Y_i) &= \mathrm{TF}(g, Y_i) \times \mathrm{IDF}(g, Y_i) \\ &= \frac{f_g}{\sum_{k\in Y_i} f_k} \times \log\frac{N'}{N_g},\end{aligned} \quad (5)$$

Here, f denotes the count. N' represents the size of the candidate set. N_g refers to the number of samples in which the n-gram g appears.

Subsequently, the diversity score of a sample $\{X_i, Y_i\}$ is:

$$S_{\mathrm{DIV}}^{i} = \sum_{g\in Y_i} \alpha_g \times \text{TF-IDF}(g, Y_i). \quad (6)$$

α is the weight newly introduced to measure the importance of the n-gram, which is initialized to 1. The higher the diversity score, the more informative the response is.

To select a diverse subset, we employ a greedy approach by dynamically adjusting the weight α. Specifically, we reduce the weight of the n-grams in the selected samples at step t by:

$$\alpha_g^{t+1} = b \times \alpha_g^t, \quad (7)$$

where $0 \le b < 1$. In this way, samples covering diverse contents reflected by different lexical features will be selected with an increased possibility.

2.3. Iterative Data Selection Algorithm

Building upon the complexity and diversity scores defined earlier, we introduce an iterative data selection algorithm, dubbed ITERIT, designed to effectively integrate both metrics for the purpose of identifying high-quality samples. Our approach adheres to the complexity-first and diversity-aware principle, which views the complexity—reflecting the pairwise relationship between the instruction and the response of a sample—as the foundational requirement. Simultaneously, it strives to maximize diversity among the candidates that already exhibit high complexity.

In alignment with the intuition to adapt to the evolving dynamics of the model during fine-tuning and to prevent the introduction of a substantial computational burden, we implement the complexity score in a coarse-to-fine, epoch-wise fashion. The complexity score for all samples is calculated at the beginning of the SFT process, and only the top-$(a \times M)$ samples are reserved for re-calculation and fine-grained selection after each epoch iteratively. M refers to the number of samples to be selected for each epoch and $a > 1$ is the coefficient for candidate re-calculation. Therefore, the time complexity for score calculation can be reduced from $O(\#\text{steps} \times N)$ to $O(N + a \times M \times (\#\text{epochs} - 1))$, where both $\#\text{epochs} \ll \#\text{steps}$ and $M \ll N$.

Before each training epoch, we filter out the samples with $S_{\mathrm{COM}}^{i,\theta_t} \ge 1$ among the $a \times M$ candidates, and calculate the comprehensive score for each sample as follows:

$$S^i = S_{\mathrm{COM}}^{i,\theta_t} \times S_{\mathrm{DIV}}^{i}. \quad (8)$$

The sample with the highest S^i will be selected. Subsequently, we update the S_{Div}^{i} according to Eq. 7, which will in turn impact S^i. ITERIT repeats this process greedily, until a total of M samples are collected for the instruction-tuning phase of the current epoch.

Overall, our data selection algorithm, designed to balance the dual objectives of effectiveness and efficiency in instruction tuning, is achieved in an iterative manner. The algorithm is elaborated in Algorithm 1.

Algorithm 1 The iterative data selection algorithm.

Input: the pre-trained model $\mathcal{M}_{\theta_0}$, the initial instruction tuning dataset D

Parameter: the number of epochs $t \in \{0, ..., T-1\}$, the number of samples selected in each epoch M, the re-calculation parameter a, the weight decay parameter b.

Output: the instruction-tuned model $\mathcal{M}_{\theta_T}$

for training epoch $t = 0, ..., T-1$ **do**
  // Data selection process
  Calculate $S^{i,\theta_t}_{\text{COM}}$ for all samples in D
  if t==0 **then**
    Sort samples in D based on $S^{i,\theta_t}_{\text{COM}}$
    $D = D[: a \times M]$
  end if
  $D' = \{(X_i, Y_i) \mid (X_i, Y_i) \in D, S^{i,\theta_t}_{\text{COM}} < 1\}$
  Instruction-tuning data for the current epoch $D_t = \emptyset$
  while $|D_t| < M$ **do**
    Calculate S^i_{DIV} for all samples in D'
    Calculate the comprehensive score S^i for all samples in D'
    $D_t = D_t \cup \{(X_i, Y_i) \in D' : S^i = \max_{(X_j,Y_j)\in D'} S^j\}$
    Update S^i_{DIV} according to Eq. 7
  end while
  // Training process
  for training steps in an epoch **do**
    Randomly sample a batch B from D_t
    Update M_{θ_t} by the Cross Entropy Loss.
  end for
end for
return $M_{\theta_{T-1}}$

3. Experimental Setup

3.1. Instruction-tuning Datasets

We conduct instruction tuning upon LLMs with four different instruction-tuning datasets. **Alpaca** (Taori et al., 2023) contains 52,000 samples that are created by leveraging text-davinci-003 model under the self-instruct framework (Wang et al., 2023). **Alpaca-GPT4** (Peng et al., 2023) contains higher quality responses generated by GPT-4 given the same instructions from Alpaca. **WizardLM** (Xu et al., 2023) refers to the 70K evolved samples collected by the Evol-Instruct algorithm that rewrites the initial instruction from Alpaca step by step into more complex instruction by ChatGPT. **Dolly** (Conover et al., 2023) consists of 15K human-generated prompt-response pairs for instruction tuning.

3.2. Evaluation Benchmarks

To evaluate the capabilities of instruction-tuned LLMs comprehensively, we selected widely-adopted benchmarks across a spectrum of targeted abilities. They include GSM8K (Cobbe et al., 2021) for arithmetic reasoning, MMLU (Hendrycks et al.) for factual knowledge, TruthfulQA (Lin et al., 2022) for safety, BBH (Suzgun et al., 2023) for multi-step reasoning and HumanEval (Chen et al., 2021) for coding capability, ARC (Clark et al., 2018) for scientific questions and Hellaswag (Zellers et al., 2019) for commonsense understanding. To guarantee the fairness of evaluation, all of the models are evaluated using publicly available code bases, including open-instruct [1] and Open LLM Leaderboard [2].

Moreover, we employ MixEval (Ni et al., 2024), which adeptly captures the breadth and subtlety of real-world user queries, demonstrating a 0.96 correlation with Chatbot Arena. Specifically, we conducted our evaluation using MixEval-hard-0601 to gauge the model's proficiency in handling general user queries.

3.3. Baselines

We compared our approach with five representative baselines as follows:

- **Vanilla** refers to supervised fine-tuning with the whole dataset.
- **Longest** (Zhao et al., 2024; Shen, 2024) is a rule-based method that selects the samples with the longest response.
- **Deita** (Liu et al., 2024b) trains scoring models based on labels collected from ChatGPT for complexity and quality assessments, and measures diversity via distances of model embeddings. The samples are selected greedily with the highest complexity-quality scores while not being redundant with the others.
- **Superfiltering** (Li et al., 2024b) improves the model-specific IFD score (Li et al., 2024c) and proposes to select samples with the highest IFD score.
- **GraphFilter** (Wu et al., 2024a) utilize the IFD score for complexity and TF-IDF scores based on instructions for diversity. The data are selected greedily by the priority score defined as the multiplication of both metrics.

For fair comparisons, we calculate the IFD scores using the target model itself instead of a smaller model such as GPT-2 (Radford et al., 2019). Besides, the same number of samples are selected by different approaches and all of the models are updated for the same steps during fine-tuning.

3.4. Implementation Details

We mainly carried out experiments on the LLaMA-3-8B pre-trained language model (Dubey et al., 2024). All of the models are trained with batch size equaling 32 for 3 epochs. The learning rate is set to 2e-5 following the training

[1] https://github.com/allenai/open-instruct
[2] https://github.com/EleutherAI/lm-evaluation-harness/

Table 1. Performance of approaches backed on Llama-3-8B fine-tuned with general instruction-following datasets. The highest score in each column is in bold, and the second-best ones for overall performance are underlined. ⋆ marks our proposed approach. All of the experiments are re-implemented upon the same pre-trained model for fair comparisons.

Method	GSM8K	MMLU	Truthful QA	BBH	Human Eval	ARC	Hella Swag	AVG	MixEval
Results on Alpaca									
Vanilla	20.00	52.20	42.78	48.52	28.05	58.53	81.77	47.41	27.80
Longest	56.50	59.12	43.12	57.59	42.80	62.20	**83.58**	<u>57.84</u>	**35.60**
Deita	37.00	56.11	42.20	52.69	38.66	64.76	83.00	53.49	29.05
Superfiltering	51.50	52.09	**43.72**	55.19	38.54	62.03	83.37	55.21	31.20
GraphFilter	44.50	**59.59**	43.57	57.31	42.32	**65.19**	83.12	56.51	29.35
ITERIT⋆	**61.00**	59.31	40.81	**59.44**	**46.10**	61.43	83.11	**58.74**	**35.60**
Results on Alpaca-GPT4									
Vanilla	41.00	50.78	52.69	48.61	39.76	59.98	82.42	53.61	34.15
Longest	60.60	59.93	54.61	56.67	42.20	61.69	**84.18**	59.97	**40.60**
Deita	56.50	56.39	52.56	51.48	37.80	63.14	84.09	57.42	33.10
Superfiltering	59.00	**60.55**	56.19	57.04	44.27	62.80	83.84	60.53	36.35
GraphFilter	62.50	58.62	**58.58**	57.22	**46.22**	**63.23**	83.81	<u>61.45</u>	33.00
ITERIT⋆	**68.50**	60.35	56.93	**60.37**	44.63	60.67	83.92	**62.20**	<u>40.15</u>
Results on WizardLM									
Vanilla	56.00	53.15	49.68	53.43	45.12	57.17	81.74	56.61	35.30
Longest	60.00	59.88	48.43	57.78	**49.39**	59.81	83.09	59.75	<u>36.55</u>
Deita	62.00	56.68	47.28	54.17	48.45	61.43	83.19	59.03	31.10
Superfiltering	56.50	59.17	**50.97**	59.81	47.68	61.26	83.00	<u>59.77</u>	34.50
GraphFilter	58.50	**60.79**	47.06	**60.56**	46.22	61.35	83.02	59.64	33.00
ITERIT⋆	**62.00**	60.33	50.73	58.80	45.24	**61.69**	**83.27**	**60.29**	**37.20**

configuration in the Alpaca codebase [3]. ITERIT selects only 5% of the whole training set for fine-tuning, i.e., $M = 0.05 * N$. We set $a = 3$ for fine-grained candidate preparation. b equals 0.1 for Alpaca, WizardLM and Dolly, and 0.0 for Alpaca-GPT-4. Other setup variations are specified in the following sections.

4. Results

We first present the overall comparisons with competitive baselines on different instruction-following datasets, followed by generalization performance on domain-specific datasets and other backbone models.

4.1. Performance on General Instruction-following Datasets

We compared models trained with different data selection approaches on various datasets. The results on Dolly are in Appendix A due to space limitation.

Results on individual benchmarks As shown in Table 1, none of the approaches consistently outperform the others among different benchmarks. Superior scores on one or two datasets also don't necessarily lead to the best overall performance. For instance, GraphFilter achieved leading performance with 59.59% and 65.19% accuracy on MMLU and ARC, respectively. Nevertheless, its overall performance, as reflected by the average score and MixEval, falls outside the top-2 approaches. In other words, data selection methods can easily achieve significant improvement in a single task when the selected data is closer to the distribution of a specific group of test data. Therefore, the overall performance of the model is more important for evaluating the effectiveness of the method, and single-task performance can be regarded as an indicator of detailed abilities.

Results on overall performance According to the average scores across the seven benchmarks and the MixEval benchmark, we have the following observations:

Previous data selection methods struggle to achieve consistent improvements over Vanilla among different datasets. Deita outperforms Vanilla by 1.25% on MixEval trained on the Alpaca dataset, while lags behind Vanilla by 1.05% and 4.2% on Alpaca-GPT4 and WizardLM correspondingly, where the instruction-response pairs are of better quality.

[3] https://github.com/tatsu-lab/stanford_alpaca

Table 2. Performance of approaches backed on Llama-3-8B fine-tuned with CodeAlpaca.

Method	HumanEval			MBPP			AVG
	pass@1	pass@5	pass@10	pass@1	pass@5	pass@10	
Vanilla	38.35	44.41	45.73	39.71	40.85	41.01	41.68
Longest	**42.87**	49.82	52.44	36.98	44.27	46.06	45.40
Deita	35.37	41.79	44.51	38.45	42.67	43.17	40.99
Superfiltering	36.77	44.86	48.17	36.62	40.61	41.73	41.46
GraphFilter	42.26	48.04	50.00	39.71	43.37	44.24	44.60
ITERIT⋆	41.04	**49.83**	**54.27**	**40.65**	**46.56**	**48.20**	**46.76**

The same trends are reflected by Superfiltering and GraphFilter.

Longest shows superior performance compared to other baselines. Although it does not perform exceptionally well on most of the single-dimension benchmarks, it shows favorable performance in the overall metrics. Longest not only consistently outperforms Vanilla with a mere 5% of original data, but also surprisingly beats the other data selection methods that require additional computations for selecting a high-quality subset.

Our proposed ITERIT demonstrates consistent improvements over Vanilla on various instruction-tuning datasets and beats the strongest rule-based approach, Longest, on most of the metrics. It also shows stronger generalization ability than Longest, as discussed in the following sections. On the other hand, although complexity and diversity have been explored in other baselines, ITERIT further enhances the benefits of combining complexity and diversity by emphasizing the synergy between the model and the data, thereby achieving state-of-the-art performance.

4.2. Performance on Domain-Specific Data

We further investigate the effectiveness of various data selection approaches on domain-specific datasets. Specifically, we select CodeAlpaca (Chaudhary, 2023), an instruction tuning dataset collected in a manner similar to Alpaca, which is designed to improve the code generation capabilities of LLMs. We utilize the HumanEval (Chen et al., 2021) and MBPP+ (Liu et al., 2023) benchmarks, which are evaluated using the pass@k metrics, indicating the success rate of a model within k attempts.

According to the results presented in Table 2, ITERIT exhibits superior performance in code generation scenarios. It consistently outperforms Longest, with the exception of pass@1 on HumanEval. Furthermore, among all approaches, ITERIT and GraphFilter are the only two that consistently surpass Vanilla across all evaluation metrics.

4.3. Performance on Other Backbone Model

To assess the generalization capability across different backbone pre-trained models, we conducted additional experiments on Qwen-2.5-7B (Team, 2024), and compared ITERIT with Vanilla and the strongest baseline, Longest. The overall results are listed in Table 3, with more details elucidated in Appendix B. ITERIT consistently outperforms both Longest on most metrics and Vanilla.

Table 3. Performance of Approaches backed on Qwen-2.5-7B fine-tuned with general instruction-following datasets.

Dataset	Benchmark	Vanilla	Longest	ITERIT
Alpaca	AVG	65.17	**66.49**	66.43
	MixEval	38.30	38.45	**38.75**
Alpaca -GPT4	AVG	68.96	68.66	**69.04**
	MixEval	40.15	39.50	**40.95**
Wizard LM	AVG	69.17	69.05	**69.32**
	MixEval	38.20	39.50	**40.50**

5. Ablations and Analysis

In this section, we analyze the rationale behind the design of ITERIT with ablation studies and data visualization. First, we show the effectiveness of iterative selection. Following that, we conduct ablations to establish the significance of incorporating diversity in our complexity-first data selection algorithm. Next, we analyze the characteristics of the selected data. Lastly, we assess the sensitivity of hyperparameters newly introduced in ITERIT.

5.1. Ablation for Iterative Selection

We verify the necessity of performing iterative selection by comparing the performance of our complete approach (**w-iteration**) with the ablation (**wo-iteration**), which uses a fixed subset selected at the beginning of the fine-tuning process. Figure 2 indicates that the average performance over 7 benchmarks consistently drops when removing the iterative selection operation, demonstrating the importance of catering to the dynamic changes of the model by updating

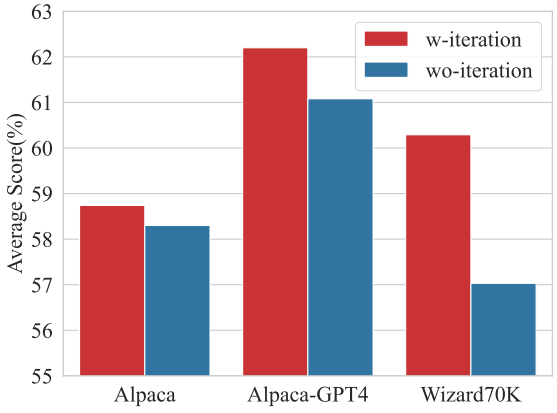


Figure 2. Ablation on the need of iterative selection. Models are evaluated by the average performance(%) over 7 datasets.

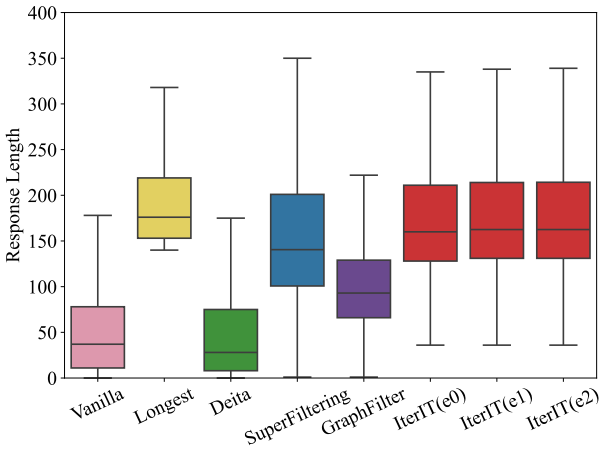


Figure 3. The response length of instruction-tuning data selected from Alpaca by different approaches.

model-specific complexity scores during the fine-tuning process.

5.2. Ablation for Introducing Diversity

To analyze the importance of incorporating the diversity measurement into the complexity-first iterative data selection algorithm, we conduct experiments on the Alpaca dataset with the following ablations:

- **"w/o div"** refers to the algorithm that does not incorporate S^i_{DIV} and selects samples based on $S^{i,\theta}_{\text{COM}}$.
- **"w div(·)"** refers the algorithm that utilizes S^i_{DIV} calculated on different part of the data. i, o, $i+o$ represents X_i, Y_i and the concatenation of (X_i, Y_i), respectively.

Results are listed in Table 4. "w/o div" lags behind most of the other ablations considering the diversity metric, indicating the significance of making a balance between complexity and diversity. Using S^i_{DIV} based solely on the instruction does not help enhance the model's overall performance, as evidenced by the lowest average score. Our approach ITERIT, i.e., "w div(o)", shows superior performances among ablations, highlighting the importance of defining the diversity of a sample based on its response. It achieves a notable improvement on GSM8K, MMLU, BBH and HumanEval compared to "w/o div".

5.3. Analysis on Selected Data

We dive deeper into analyzing the characteristics of the selected instruction tuning data by different approaches. Considering the strong performance of Longest, we first analyze the response length of different approaches measured by the number of words, as well as the three subsets selected by our ITERIT for each epoch. The statistics for Alpaca are shown in Fig. 3, with additional results in Appendix C.

The overall trend among different approaches is Longest>ITERIT>Superfiltering>GraphFilter>Vanilla>Deita, similar to the performance trend show in Table 1. Nevertheless, the reversed pairwise relations (Longest, ITERIT) and (Deita, Vanilla) show that length is not the only golden criterion for data selection. Our approach selects data based on complexity and diversity, outperforming Longest with smaller response lengths.

We further analyze the similarity between the selected subsets by different approaches. The Jaccard similarity between the subset of Longest and the ones used in each epoch by ITERIT is all around 50%. Besides, the lower quartile of ITERIT is much lower than that of Longest in Fig. C. These observations suggest that samples with shorter responses can also be valuable for instruction tuning, which is not only a key difference between the selected data but also contributes to the favorable performance of ITERIT. In addition, the similarity between the subsets in our approach is listed in Table 5. The Jaccard similarity is much lower between Epoch 1 and Epoch 3, showing the model's preference to the instruction-tuning data changes as more update steps are carried out. ITERIT made adjustments to the data selection during fine-tuning for catering to the dynamics of the model.

5.4. Sensitivity of Hyper-parameters

We discuss the effects of hyper-parameters in ITERIT, including the selected data size for each epoch, the candidate size for re-calculation controlled by a, and the weight decay coefficient b for diversity measurement.

Data Size The model's performance with selection size M set to different percentages of the original dataset is illustrated in Fig. 4(a). The performance does not exhibit a positive correlation with the amount of selected data. When

Table 4. Ablation performance for the incorporation of diversity metric. The best results are in **bold**.

Ablations	GSM8K	MMLU	TruthfulQA	BBH	HumanEval	ARC	HellaSwag	AVG
w/o div	57.5	53.90	43.01	57.31	36.83	**62.46**	83.02	56.29
w div(i)	46.00	57.53	**45.83**	56.30	41.10	61.35	82.72	55.83
w div($i+o$)	56.50	58.32	42.43	57.87	41.46	61.95	82.39	57.27
w div(o)	**61.0**	**59.31**	40.81	**59.44**	**46.10**	61.43	**83.11**	**59.74**

Table 5. Jaccard similarity(%) between pairs of subset selected by ITERIT during instruction tuning.

Epochs	1,2	2,3	1,3
Jaccard Similarity	78.88	79.43	70.55

only 1% of the data is selected, ITERIT requires a greater number of training epochs to achieve convergence, resulting in frequent recalculation processes and a reduction in training efficiency. Meanwhile, the performance does not improve even when the model is trained on more data with additional steps. At this point, the recalculation process cannot be executed in a timely manner, which hinders the synergy between the model and the data, leading to suboptimal performance. Therefore, we suggest setting M in the range of 1K to 5K considering the balance between training efficiency and the model's performance. For fair comparisons, we set $M = 0.05N$ across different datasets in this paper.

Candidate Pool Performance under different candidate sizes calculated by the multiplication of a and M for recalculation, is shown in Fig. 4(b). Enlarging the candidate size not only lifts the computational load, but also increases the likelihood of low-quality data being selected due to the imperfect complexity or diversity score. Therefore, we propose a coarse-to-fine way for updating the complexity score and suggest setting $a = 3$. In this way, low-quality samples recognized at the beginning will be removed directly, and sufficient high-quality samples will be considered for diversity consideration.

Weight Decay The rationale behind introducing weight decay in the diversity score S^i_{DIV} is illustrated in Fig. 4(c). We hypothesize that removing the TF-IDF scores for already selected n-grams may not be appropriate for mathematical data and code data. These samples contain reserved words that have a high repetition rate. Setting b to 0 reduces the likelihood of such data being selected, which may adversely affect the reasoning capabilities of LLMs. This can be further verified by the 2.50%, 2.59% and 4.76% improvement of $b = 0.1$ on GSM8K, BBH and HumanEval, respectively, compared with $b = 0.0$. On the other hand, adopting a large b will have a negative impact on the diversity of data. Therefore, we set $b = 0.1$ for most instruction-tuning scenarios involving data from various tasks, and suggest $b = 0.0$ for task-specific scenarios where reserved words are either undefined or shared across all samples.

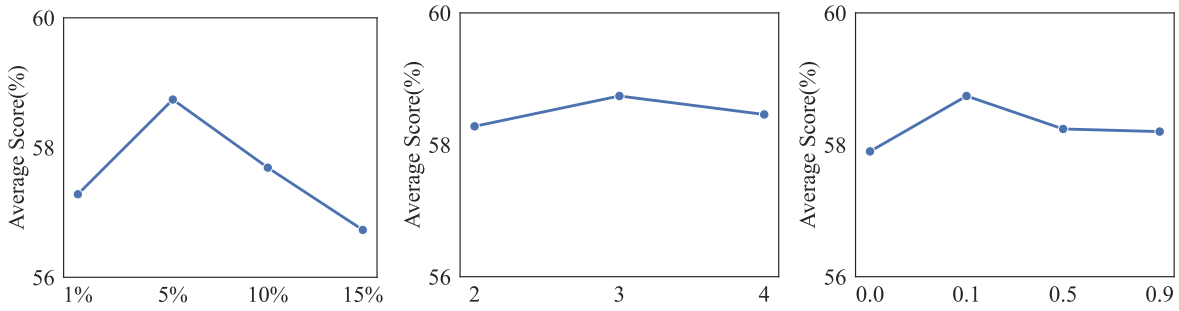
6. Related Work

6.1. Instruction Tuning Datasets

Instruction tuning teaches LLMs to perceive the intent of users and provide helpful responses, standing as a core component in the deployment of LLMs (Ouyang et al., 2022). Collecting high-quality data for instruction tuning has caught great attention. Early works (Wei et al., 2022; Longpre et al., 2023) merge existing NLP datasets to obtain a diverse collection across different tasks. Subsequently, Wang et al. (2023) proposed the Self-Instruct framework, which is an automated algorithm gathering instruction-response data by bootstrapping LLMs' generations. Building on this work, a number of data synthesis and evolution techniques (Xu et al., 2023; Ding et al., 2023; Wu et al., 2024b; Li et al., 2024a) have emerged, leveraging powerful proprietary LLMs to significantly enhance the quality and size of candidate datasets for supervised alignment. Nonetheless, according to the superficial alignment hypothesis proposed by Zhou et al. (2024), LLMs have acquired abundant knowledge and abilities during pre-training, and the focus of instruction tuning is all about style learning for providing a helpful response. They verified their hypothesis with around 1K elaborately selected samples, highlighting data selection for instruction tuning as a promising research direction.

6.2. Data Selection for Instruction Tuning

Although it's widely accepted that the quality of instruction tuning data is more significant than the quantity, the criteria for quality measurement remains mysterious. AlpaGasus (Chen et al., 2023) relies on ChatGPT's understanding and prompts it to score samples based on its quality. Cao et al. (2023) relies on a bag of indicators measuring quality from different aspects, such as length, coherence, understandability, etc., trying to estimate the models' performance trained on a selected subset. Nuggests (Li et al., 2024d) leverages one-shot learning performance to discern



(a) Selection Size (b) a for Candidate Size (c) Weight Decay b

Figure 4. The average scores(%) of models trained on Alpaca under different hyper-parameters.

the quality of the data, and SelectIT (Liu et al., 2024a) utilizes the intrinsic uncertainty of LLMs from different levels, including token, sentence and model, to make a collaborative decision. Meanwhile, research from Shen (2024) and Zhao et al. (2024) argue that selecting the longest responses is a simple but tough-to-best baseline, which aligns with the superficial alignment hypothesis. They find that the longest samples are more challenging, high-quality and preferred by humans.

More works consider complexity and diversity for assessing data quality. InsTag (Lu et al., 2023) queries proprietary models to tag samples based on semantics and intentions, and defines diversity and complexity measurements regarding the number of tags. Li et al. (2024b) measures the complexity from the aspect of instruction-following difficulty and proposes the model-specific IFD score (Li et al., 2024c). Deita (Liu et al., 2024b) trains an additional scoring model based on labels collected from ChatGPT from the aspects of complexity and quality, and considers diversity via distance between samples' semantic representations during the selection process. Wu et al. (2024a) considers lexical diversity based on bipartite graph and IFD score at the same time, outperforms a number of baselines (Wang et al., 2024; Arthur & Vassilvitskii, 2006; Li et al., 2024b) that mainly consider a single dimension. Our approach ITERIT is in line with these works and aims at advancing instruction tuning performance in an efficient way.

7. Conclusion

We introduce ITERIT for improving LLMs' instruction tuning by iterative data selection. Our approach successfully unleash the power of quality metrics by perceiving the model dynamics for complexity calculation and exploiting response informativeness for diversity calculation. By integrating both aspects, our approach demonstrates superior performance on multiple instruction tuning datasets and generalizes well to domain-specific scenarios. We underscore the importance of model-data collaboration towards powerful LLMs and will devote into more complicated scenarios and different training stages of LLMs in the future.

References

Arthur, D. and Vassilvitskii, S. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

Cao, Y., Kang, Y., Wang, C., and Sun, L. Instruction mining: Instruction data selection for tuning large language models. *arXiv preprint arXiv:2307.06290*, 2023.

Chaudhary, S. Code alpaca: An instruction-following llama model for code generation. *GitHub repository*, 2023.

Chen, L., Li, S., Yan, J., Wang, H., Gunaratna, K., Yadav, V., Tang, Z., Srinivasan, V., Zhou, T., Huang, H., et al. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*, 2023.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. D. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Conover, M., Hayes, M., Mathur, A., Xie, J., Wan, J., Shah, S., Ghodsi, A., Wendell, P., Zaharia, M., and Xin, R. Free dolly: Introducing the world's first truly open instruction-tuned llm. *Company Blog of Databricks*, 2023.

Ding, N., Chen, Y., Xu, B., Qin, Y., Hu, S., Liu, Z., Sun, M., and Zhou, B. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3029–3051, 2023.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Lee, C., Roy, R., Xu, M., Raiman, J., Shoeybi, M., Catanzaro, B., and Ping, W. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*, 2024.

Li, M., Chen, L., Chen, J., He, S., Gu, J., and Zhou, T. Selective reflection-tuning: Student-selected data recycling for llm instruction-tuning. *arXiv preprint arXiv:2402.10110*, 2024a.

Li, M., Zhang, Y., He, S., Li, Z., Zhao, H., Wang, J., Cheng, N., and Zhou, T. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14255–14273, 2024b. doi: 10.18653/v1/2024.acl-long.769.

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

Li, Y., Hui, B., Xia, X., Yang, J., Yang, M., Zhang, L., Si, S., Chen, L.-H., Liu, J., Liu, T., Huang, F., and Li, Y. One-shot learning as instruction data prospector for large language models. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4586–4601, 2024d. doi: 10.18653/v1/2024.acl-long.252.

Lin, S., Hilton, J., and Evans, O. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, 2022.

Liu, J., Xia, C. S., Wang, Y., and Zhang, L. Is your code generated by chatGPT really correct? rigorous evaluation of large language models for code generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=1qvx610Cu7.

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

Liu, W., Zeng, W., He, K., Jiang, Y., and He, J. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024b.

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

Lu, K., Yuan, H., Yuan, Z., Lin, R., Lin, J., Tan, C., Zhou, C., and Zhou, J. # instag: Instruction tagging for analyzing supervised fine-tuning of large language models. In *The Twelfth International Conference on Learning Representations*, 2023.

Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*, 2022. doi: 10.48550/ARXIV.2210.07316. URL https://arxiv.org/abs/2210.07316.

Ni, J., Xue, F., Yue, X., Deng, Y., Shah, M., Jain, K., Neubig, G., and You, Y. Mixeval: Deriving wisdom of the crowd from llm benchmark mixtures. *arXiv preprint arXiv:2406.06565*, 2024.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Peng, B., Li, C., He, P., Galley, M., and Gao, J. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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

Suzgun, M., Scales, N., Schärli, N., Gehrmann, S., Tay, Y., Chung, H. W., Chowdhery, A., Le, Q., Chi, E., Zhou, D., et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Team, Q. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Wang, H., Xiong, W., Xie, T., Zhao, H., and Zhang, T. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*, 2024.

Wang, Y., Mishra, S., Alipoormolabashi, P., Kordi, Y., Mirzaei, A., Arunkumar, A., Ashok, A., Dhanasekaran, A. S., Naik, A., Stap, D., et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*, 2022.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. 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)*, pp. 13484–13508, 2023.

Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.

Wu, M., Vu, T.-T., Qu, L., and Haffari, G. The best of both worlds: Bridging quality and diversity in data selection with bipartite graph. *arXiv preprint arXiv:2410.12458*, 2024a.

Wu, M., Waheed, A., Zhang, C., Abdul-Mageed, M., and Aji, A. Lamini-lm: A diverse herd of distilled models from large-scale instructions. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 944–964, 2024b.

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

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

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

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

A. Performance on Dolly Dataset

The performance of Dolly is shown in Table 6. All of the approaches struggle to make improvements in all aspects. GraphFilter achieves superior scores on 4 out of 7 benchmarks, with a compromise on math reasoning ability, leading to its inferior overall performance. Longest and ITERIT ranks top 1 and 2 according to the average score, respectively, with ITERIT topping the rank on MixEval.

B. Performance on Qwen-2.5-7B

Detailed performance on Qwen-2.5-7B is in Table 7. Our approach ITERIT's overall performance consistently outperforms Vanilla, and ranks toppest on 5 out of 6 cases.

C. Analysis of Selected Data

The statistics of response lengths for instruction-tuning data selected from Alpaca-GPT4 and WizardLM by different approaches are depicted in Figure 5. The average response length of data chosen by our ITERIT is longer than the other baselines, while covering data with short responses compared to Longest.

The Jaccard similarity between the subsets selected before each epoch on Alpaca-GPT4 and WizardLM is in Table 8. The trends are identical to that of Alpaca.

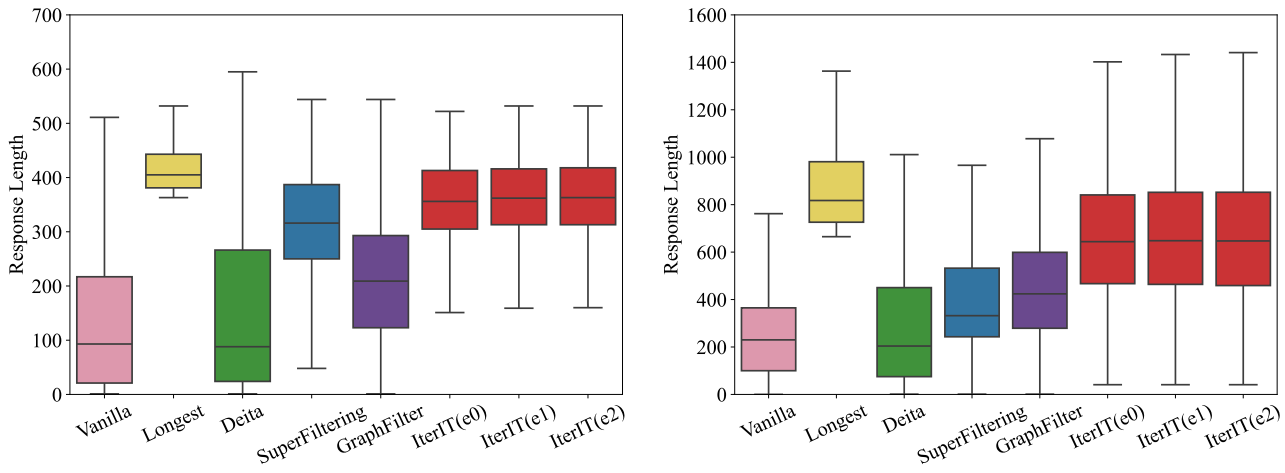
We further extract the response representations of samples by NV-Embed (Lee et al., 2024), which achieves the state-of-the-art performance on MTEB benchmark (Muennighoff et al., 2022). The scatter plot visualized using t-SNE on response representations from the Alpaca dataset is shown in Figure 6. Deita and GraphFilter select data with better semantic diversity, while both of them don't reflect superior performance in Table 1. ITERIT shares some semantic distribution similarities with Longest. Nonetheless, the specific data points selected by both approaches are different as analyzed in Sec. 5.3. ITERIT shows favorable performance and strong generalization ability, demonstrating its successful design of combining the complexity and diversity metrics based on the instruction-following difficulty and response informativeness, respectively.

Table 6. Performance of approaches backed on Llama-3-8B fine-tuned with Dolly. The highest score in each column is in bold, and the second-best ones for overall performance are underlined. ⋆ marks our proposed approach.

Method	GSM8K	MMLU	Truthful QA	BBH	Human Eval	ARC	Hella Swag	AVG	MixEval
Results on Dolly									
Vanilla	49.5	56.14	42.51	55.65	42.32	60.75	83.78	55.81	28.70
Longest	63.00	**61.09**	42.88	59.54	41.34	61.43	84.72	**59.14**	32.55
Deita	48.00	57.04	37.01	53.33	42.80	63.05	84.22	55.06	28.85
Superfiltering	57.50	60.13	42.52	58.43	40.97	**63.57**	84.81	58.28	31.50
GraphFilter	53.00	60.77	**43.67**	**59.63**	**44.76**	61.86	**84.89**	58.36	<u>32.85</u>
ITERIT⋆	**65.00**	60.70	41.92	56.20	41.71	61.60	84.64	<u>58.82</u>	**33.80**

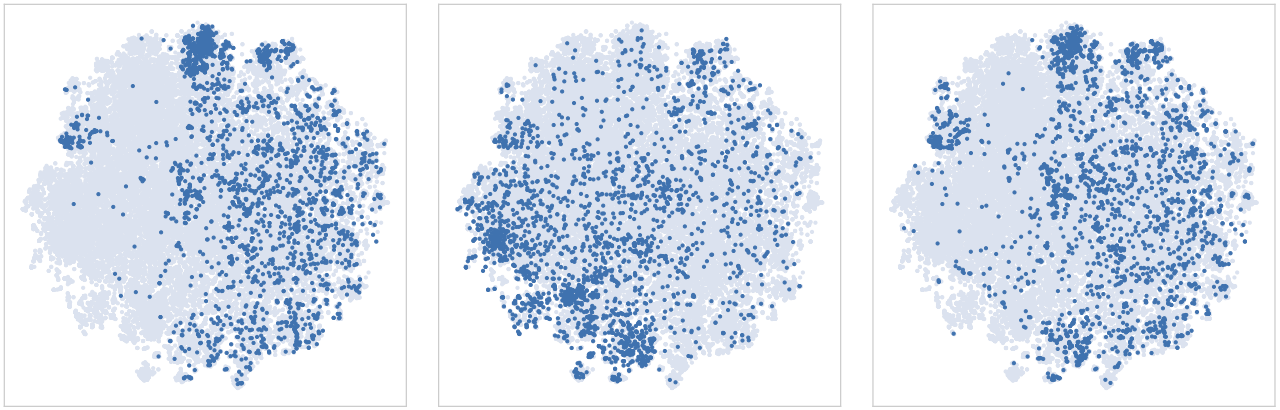
Table 7. Performance of approaches backed on Qwen-2.5-7B.

Method	GSM8K	MMLU	Truthful QA	BBH	Human Eval	ARC	Hella Swag	AVG	MixEval
Results on Alpaca									
Vanilla	74.00	69.50	**46.50**	**66.94**	54.27	**64.85**	**80.14**	65.17	38.30
Longest	**86.00**	70.74	44.95	66.20	53.78	63.91	79.90	**66.49**	<u>38.45</u>
ITERIT⋆	84.00	**70.77**	45.32	**66.94**	**54.76**	63.31	79.94	<u>66.43</u>	**38.75**
Results on Alpaca-GPT4									
Vanilla	85.50	70.91	55.30	66.30	62.07	63.23	**80.85**	<u>69.17</u>	38.20
Longest	87.00	71.38	56.75	65.28	**58.78**	63.57	80.59	69.05	<u>39.50</u>
ITERIT⋆	**87.50**	**71.41**	**57.22**	**66.67**	57.93	**63.82**	80.68	**69.32**	**40.50**
Results on WizardLM									
Vanilla	**88.50**	70.62	**59.44**	**65.37**	56.10	62.20	80.51	<u>68.96</u>	<u>40.15</u>
Longest	87.50	**71.31**	56.46	62.69	56.59	64.68	81.36	68.66	39.50
ITERIT⋆	88.00	71.14	57.27	63.43	**56.95**	**65.02**	**81.50**	**69.04**	**40.95**

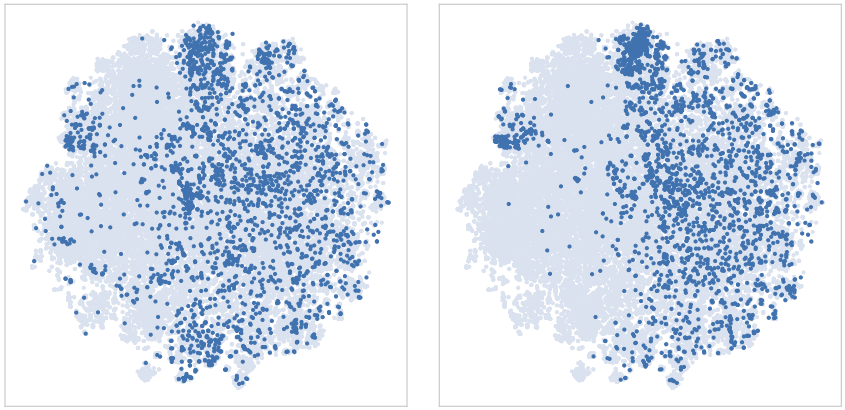


(a) Alpaca-GPT4 (b) WizardLM

Figure 5. Response lengths of instruction-tuning data selected from Alpaca-GPT4 and WizardLM.



(a) Longest (b) Deita (c) Superfiltering



(d) GraphFilter (e) ITERIT

Figure 6. Visualization using t-SNE of data points from Alpaca dataset. Grey points represent samples from the dataset, and blue ones represent samples selected by corresponding approaches.

Table 8. Jaccard similarity(%) between pairs of subset selected by ITERIT during instruction tuning.

Epochs	Alpaca-GPT4	WizardLM
1,2	81.06	83.29
2,3	82.20	85.48
1,3	73.04	77.48