# Self-Evolved Diverse Data Sampling for Efficient Instruction Tuning

**Anonymous ACL submission**

## Abstract

Enhancing the instruction-following ability of Large Language Models (LLMs) primarily demands substantial instruction-tuning datasets. However, the sheer volume of these imposes a considerable computational burden and annotation cost. To investigate a label-efficient instruction tuning method that allows the model itself to actively sample subsets that are equally or even more effective, we introduce a self-evolving mechanism DIVERSEEVOL. In this process, a model iteratively augments its training subset to refine its own performance, without requiring any intervention from humans or more advanced LLMs. The key to our data sampling technique lies in the enhancement of diversity in the chosen subsets, as the model selects new data points most distinct from any existing ones according to its current embedding space. Extensive experiments across three datasets and benchmarks demonstrate the effectiveness of DIVERSEEVOL. Our models, trained on less than 4% of the original dataset, maintain or improve performance compared with finetuning on full data. We also provide empirical evidence to analyze the importance of diversity in instruction data and the iterative scheme as opposed to one-time sampling. Our code will be made publicly available. [1]

## 1 Introduction

Large Language Models (LLMs) have demonstrated prowess in producing human-aligned response to varied instructions. A pivotal technique for enhancing the instruction-following capabilities of LLMs is Instruction Tuning, which aligns the model with human preferences using data in the form of instruction-response pairs.

While massive instruction-tuning datasets exist, their vast quantity poses a significant computational burden, and their curation is itself a formidable challenge, given the meticulous labor involved in annotations. Recent works shed light on data distillation, achieving similar or even better alignment performance relying on fewer instruction data, by mining compact subsets from extensive instruction datasets (Zhou et al., 2023; Cao et al., 2023; Chen et al., 2023). However, these works demand tremendous supervision from humans or advanced LLMs, such as GPT4 (OpenAI, 2023), for selecting the ideal subset.

In contrast, our work introduces DIVERSEEVOL, a novel method featuring a **self-evolving** mechanism. In parallel to the approach in Li et al. (2023), DIVERSEEVOL employs an iterative strategy, where the model relies on its current embedding space to augment its own training data samples that lead to an improved model in the next step. As such, instead of seeking external oversight, DIVERSEEVOL facilitates the model's **self-evolution**, as it actively selects data to refine its own performance through iterations.

Central to DIVERSEEVOL's design of data selection is the maintenance of high diversity. When curating a subset from a vast dataset, the key challenge is to ensure that this subset is as representative as possible. This indicates that data points within the subset must be diverse in order to ensure comprehensive coverage and simulate the effect of the entire dataset. Therefore, DIVERSEEVOL adopts a $K$-Center-based (Sener and Savarese, 2017) strategy that chooses data points characterized by the highest distance from any existing labeled data.

Our experiments span three distinguished instruction-tuning datasets curated by both human-annotation (Conover et al., 2023), and Self-Instruct (Taori et al., 2023; Peng et al., 2023). Consistently, through DIVERSEEVOL, our models, trained on less than 4% of the original datasets, match or outperform baselines trained on the entirety of the source datasets across all benchmarks.

Furthermore, our investigation yields two cru-

[1] See the *Software* package accompanying this submission.

cial findings. First, training dataset diversity is paramount for the success of instruction tuning. Our method's emphasis on diversity, quantified via the Vendi Score (Friedman and Dieng, 2022), correlates with enhanced model performance. Second, an iterative, evolving data sampling strategy outperforms direct, one-shot sampling. This evolution-driven approach, characterized by progressive data selection based on the model's current state, offers superior training outcomes.

In sum, our main contributions are three-fold:

- A self-evolving, efficient data sampling pipeline, DIVERSEEVOL that requires significantly less data yet matches or surpasses the performance of models trained on complete datasets.
- A quantified demonstration of the essential role of dataset diversity in instruction-tuning, emphasizing the link between training data diversity and model performance.
- A revelation that iterative, evolving sampling outperforms static, one-time sampling, underscoring the advantages of progressive data selection for model improvement.

## 2 Related Works

**Instruction Tuning and Its Efficiency.** Instruction tuning is paramount for boosting the instruction-following capabilities of LLMs, and a range of methods have been utilized to curate large-scale datasets, extending from human annotations (Conover et al., 2023; Köpf et al., 2023) to distillations from parent LLMs, such as Text-Davinci-003 (Taori et al., 2023), GPT-3.5-TURBO (Xu et al., 2023a), and GPT4 (Peng et al., 2023). The Vicuna dataset (Chiang et al., 2023), originating from ShareGPT's real-world interactions, serves as another exemplar in this regard. As the field advances, there's a growing inclination toward refining instruction tuning methods for better efficiency. AlShikh et al. (2023) shows that the instruction-tone is learned rather early without the need of training on full-sized dataset. Zhou et al. (2023) yields promising results with only 1,000 manually curated instruction data. Concurrently, leveraging advanced LLMs for instruction data labeling has emerged as a trend, with endeavors like Chen et al. (2023) using ChatGPT for data rating and filtration, and others like Lu et al. (2023) exploring diverse sampling based on open-world tag annotations. However, DIVERSEEVOL conducts diverse sampling with only its own supervision by a self-evolving mechanism while above methods necessitate external supervision from either humans and more advanced LLMs.

**Data Sampling Strategies.** Our work also draws inspirations from data-centric AI principles, emphasizing self-automated sampling strategies. These methodologies largely fall into two categories: (1) *Uncertainty*-based approaches that prioritize datapoints the model's prediction deems ambiguous. Measures of the predictive uncertainty include maximum entropy (Entropy-Sampling, Shannon, 2001), lowest logits (Least-Confidence, Wang and Shang, 2014), and minimal differences in the likelihood of top two probable labels (Margin-Sampling, Netzer et al., 2011). (2) *Diversity*-based approaches that focus on a representative subset within the model's embedding space. Such strategies like $K$-Center-Sampling (Sener and Savarese, 2017) and Cluster-Margin (Citovsky et al., 2021) have gained prominence. In this work, we actively experiment above sampling strategies and empirically show that diversity-based sampling benefits the reduction of instruction data the most without harming model performance.

## 3 DIVERSEEVOL

In this section, we introduce DIVERSEEVOL, a self-evolved diverse sampling method for the +selection of instruction data. We first introduce instruction data selection as an iterative process (§3.1). Then, we lay out details about our $K$-Center-based algorithm for the selection of training data (§3.2). The overall workflow is illustrated in Fig. 1.

### 3.1 Iterative Instruction Data Selection

Our objective is to formalize instruction data mining as an iterative process, extracting from a vast source instruction dataset progressively according to a strategy. Given a collection of instruction-response pairs, denoted as $\mathcal{Z} = \{(x_i, y_i)\}_{i \in \mathbb{N}}$, where each $(x_i, y_i)$ represents a specific instruction-response pair, we define $\mathbb{N} = \{1, \ldots, n\}$ as the size of the initial source instruction dataset. The iterative procedure revolves around two data containers: the training data pool $P_t$ up to iteration step $t$ and the container of unselected data points, $Q_t$. At each iteration $t$, a selection function (i.e., strategy) $A$ determines which data points, $\mathcal{S} = \{s_j\}_{j \in \mathbb{K}}$, with $\mathbb{K} = \{1, \ldots, k\}$, are integrated into the training data pool $P_{t+1}$ for the next step. This expanded

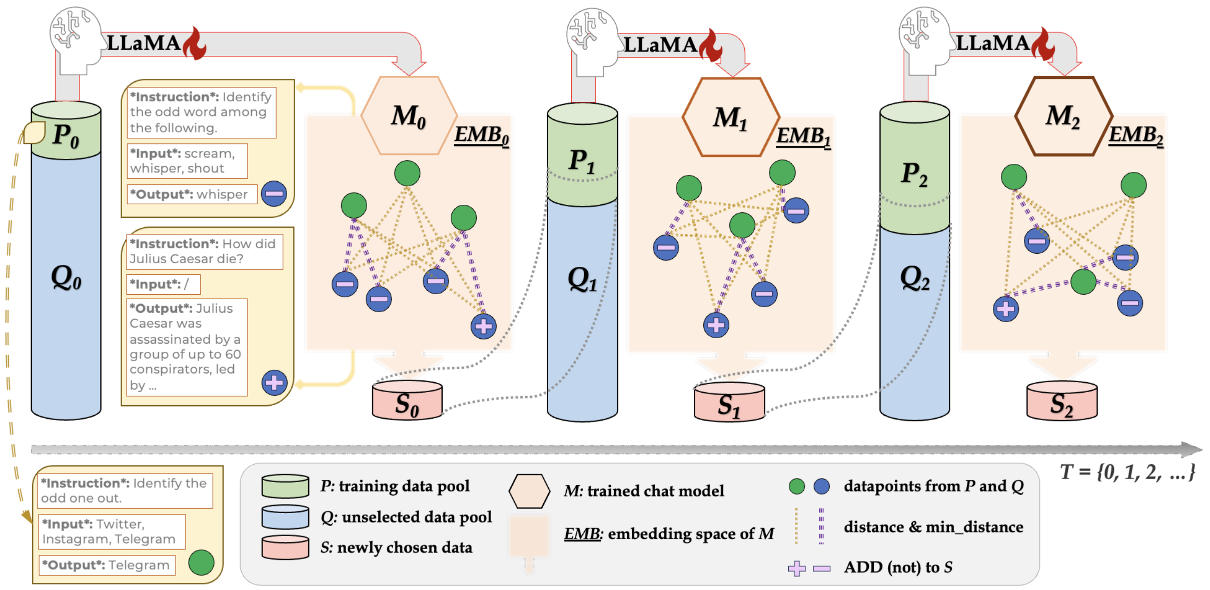


Figure 1: Overview of our iterative DIVERSEEVOL: Starting with an initial training data pool $P_0$ and the remaining data $Q_0$ from the source dataset, we train a chat model $M_0$ and project all datapoints into its embedding space $\underline{EMB_0}$. Leverage $K$-Center based selection §3.2 in this embedding space, a new set of datapoints $S_0$ is chosen from $Q_0$ and added to the next training data pool $P_1$ to instrution-tune the next chat model $M_1$. This process is repeated for $T$ steps, producing progressively augmented training data pool based solely on the model itself, which is then used to improve a more refined model with improved capabilities.

pool then serves as the training set for the next model iteration, $M_{t+1}$.

Beginning with a randomized data pool, $P_0$, to train the initial model $M_0$, every subsequent step employs model $M_t$, the current training pool $P_t$, and the comprehensive dataset $\mathcal{Z}$ to inform function $A$, which then outputs new data points $S_t$ to be added to the training pool for the next iteration $P_{t+1}$, as in: $S_t = A(\mathcal{Z}, P_t, M_t); P_{t+1} = P_t \cup S_t$. Thus, each iteration consists of two operations: 1. Deduce new data points $S_t$ to merge into $P_{t+1}$, informed by the previously trained model $M_t$. 2. Train the subsequent chat model, $M_{t+1}$, with the updated data pool $P_{t+1}$.

The efficacy of this approach hinges on the selection function $A$ that determines the additional $k$ data points for each training iteration. As $P$ grows both in volume and, crucially, in diversity (as stressed by our method, see §3.2), the resulting chat model continuously refines its capabilities.

### 3.2 Selection Algorithm: *K*-Center-Sampling

Central to DIVERSEEVOL is our selection function $A$ based on the $K$-Center-Sampling method (Sener and Savarese, 2017), as detailed in Alg. 1. The selected subset must aptly represent the broader dataset to ensure that models trained on reduced subsets rival those trained on the complete dataset. Thus, our function $A$ strives to amass a highly diverse subset of the source dataset, reminiscent of the facility location problem (Wolf, 2011; Wei et al., 2013).

With a given set of training data points, $P_t$, function $A$ identifies novel data points $S_t$ that, when combined with $P_t$, provide a representative sample of the source dataset. This entails selecting newly added data that is as **different** as possible from any of the existing data points. The "difference" from existing data points is quantified by the closest distance of a candidate datapoint (i.e., an as-yet unchosen data point from $Q_t$) to any existing training data in $P_t$. In other words: the distance to its nearest neighboring datapoint $P_t$. Therefore, our objective for $A$ at iteration $t$ can be succinctly articulated as:

**Objective:** *From a candidate pool, choose $k$ data points in such a way that the distances to their respective nearest existing training data points are maximized.*

$$\max \sum_{1 \leq i \leq k} \min_{j \in P_t} \Delta\left(\mathbf{s}_i, \mathbf{p}_j\right) \quad (1)$$

Our function aims to designate each of the $k$ new data points as a unique center within the full training pool. Consequently, it seeks to maximize the minimum distance from each new data point in $S_t$ to any existing training data point in $P_t$. As formulated below, for $k$ data points to be selected from the candidate datapoint pool $Q_t$, we select:

$$\arg\max_{i \in Q_t} \min_{j \in P_t} \Delta\left(\mathbf{s}_i, \mathbf{p}_j\right) \tag{2}$$

The embeddings produced by the currently trained model $M_t$ guide our selection since the distance between samples, denoted as $\Delta$, is computed based on the output hidden states of $M_t$ after average pooling over all token positions, which provides a more suitable embedding space for existing data. As such, data points added to the training set ensure to best supplement the existing dataset according to the model's current understanding. This iterative procedure facilitates the model's **evolution**, as it incorporates insights from prior iterations to refine its performance.

**Algorithm 1:** Iterative $K$-Center-Sampling for $T$ Steps

**Input:** $Z$: entire source dataset; $M_{pretrain}$: foundation LLM; $k$: budget for new data points; $T$: total number of iterations
**Output:** *Series* $P = \{P_0, P_1, \ldots, P_T\}$; *Series* $M = \{M_0, M_1, \ldots, M_T\}$
**Initialize:** $P_0$: $k$ data points randomly sampled from $Z$; $Q_0 = Z \setminus P_0$
**for** $t = 0$ **to** $T-1$ **do**
  **Finetune:** $M_{pretrain}$ using $P_t$ to get $M_t$
  **Select data points:**
    **initialize:** $S_t = \emptyset$; $Q'_t = Q_t$
    **repeat**
      $s = \arg\max_{i \in Q'_t} \min_{j \in P_t} \Delta\left(\mathbf{s}_i, \mathbf{p}_j\right)$
      $S_t = S_t \cup \{s\}$
      $Q'_t = Q'_t \setminus \{s\}$
    **until** $|S_t| = k$;
  **Update Pools:**
    $P_{t+1} = P_t \cup S_t$
    $Q_{t+1} = Z \setminus P_{t+1}$
**return** *Series P, Series M*

## 4 Experiments

In this section, we introduce the experimental setup (§4.1), main results (§4.2), and conduct rich analyses about the effectiveness of DiverseEvol that can be attributed to its central designs of data diversity and iterative sampling (§4.3).

### 4.1 Experimental Setup

**Datasets.** Three prominent open-source instruction-tuning datasets serve to validate the effectiveness of DiverseEvol. These include both human-annotated data (Databricks-Dolly, Conover et al., 2023) and machine-generated (SelfInstruct-Davinci, Taori et al., 2023, SelfInstruct-GPT4, Peng et al., 2023). Statistics are detailed in Tab. 2.

**Baselines.** As a data sampling method, we introduce strong baselines that correspond to chat models directly trained on the full-sized source datasets, including LLaMA-7B (Touvron et al., 2023) finetuned on Databricks-Dolly, SelfInstruct-Davinci, and SelfInstruct-GPT4 respectively. For comparison, our $K$-Center-based method, which prioritizes diversity, is also benchmarked against the following: (1) Random-Sampling: stochastically selects data points at each iteration. (2) Least-Confidence (Culotta and McCallum, 2005): samples data points the current model exhibits least confidence in, measured by the average max-logit value across the predicted token sequence. (3) Margin-Sampling (Netzer et al., 2011): chooses data points whose logits obtained by current model show minimal differences in the likelihood of top two probable tokens.

**Benchmarks.** We test our method on three distinct benchmarks: Vicuna-Bench (Chiang et al., 2023), Koala-Bench (Geng et al., 2023), and Wizardlm-Bench (Xu et al., 2023b) to ensure a extensive evaluation and help minimize test set biases. Alongside these, we adopt an evaluation framework, as in prior works (Chiang et al., 2023; Dubois et al., 2023; Zheng et al., 2023; Xu et al., 2023a), with GPT4-Judge ($J$) scoring two model responses (template detailed in Appendix A). We also randomly permute the order of the two answers to counteract potential position biases in GPT4's judgement. Specifically, we compare the answers of all chat models ($A^{\text{model}}$) to those generated by GPT3.5-TURBO ($A^{\text{chatgpt}}$), a general competitor. We then compute Relative Score (RS) and Win-And-Tie-Rate (WTR) vs. ChatGPT as metrics to assess instruction-following capabilities.

- **Relative Score (RS)** vs. ChatGPT: Compares the chat model's performance with ChatGPT based on their scores, formulated as:

$$\text{RS} = \frac{\sum_{q \in \text{testset}} J(A_q^{\text{model}})}{\sum_{q \in \text{testset}} J(A_q^{\text{chatgpt}})} \tag{3}$$

| Sampling Strategy | Vicuna-Bench | | | Koala-Bench | | | Wizardlm-Bench | | |
|---|---|---|---|---|---|---|---|---|---|
| | *RS* | *WTR* | $N_{best}$ | *RS* | *WTR* | $N_{best}$ | *RS* | *WTR* | $N_{best}$ |
| *Source Dataset = Databricks-Dolly-15K* | | | | | | | | | |
| `*Full Data` | 73.84 | 5.00 | 15011 | 57.90 | 3.33 | 15011 | 58.73 | 3.21 | 15011 |
| Random | 73.06 | 6.25# | 700 | 53.11 | 3.33* | 900 | 56.02 | 4.59* | 1100 |
| Least-Confidence | 46.68 | 0.00 | 100 | 36.01 | 2.27* | 1100 | 40.08 | 1.38 | 800 |
| Margin-Sampling | 69.67 | 3.75 | 400 | 52.29 | 5.00 | 600 | 53.53 | 3.21* | 900 |
| ***K*-Center (DIVERSEEVOL)** | **79.69** | **20.00** | 700 | **62.29** | **6.67** | 1100 | **62.94** | **8.26** | 700 |
| *Source Dataset = SelfInstruct-Davinci-52K* | | | | | | | | | |
| `*Full Data` | 73.03 | 2.50 | 52002 | **69.50** | 3.89 | 52002 | 61.59 | 5.05 | 52002 |
| Random | 75.43 | **7.50*** | 800 | 62.33 | 5.56 | 900 | 58.60 | 5.96* | 500 |
| Least-Confidence | 64.27 | 2.50 | 600 | 43.27 | 3.33# | 100 | 49.26 | 5.05* | 500 |
| Margin-Sampling | 68.98 | 2.50* | 1000 | 55.22 | 2.78 | 1000 | 53.98 | 2.75 | 1000 |
| ***K*-Center (DIVERSEEVOL)** | **79.16** | **7.50*** | 1000 | 66.95 | **6.11*** | 1100 | **63.08** | **7.80*** | 700 |
| *Source Dataset = SelfInstruct-GPT4-52K* | | | | | | | | | |
| `*Full Data` | 90.28 | 46.25 | 52002 | **80.33** | 10.56 | 52002 | **75.00** | 12.84 | 52002 |
| Random | 90.21 | 48.75# | 500 | 77.31 | 12.78 | 800 | 71.95 | **14.68*** | 1000 |
| Least-Confidence | 79.11 | 17.5* | 1100 | 55.57 | 4.44# | 800 | 58.33 | 6.88 | 100 |
| Margin-Sampling | 82.43 | 33.75# | 600 | 63.10 | 7.22 | 1000 | 65.01 | 8.26 | 1000 |
| ***K*-Center (DIVERSEEVOL)** | **91.69** | **50.00#** | 400 | 79.01 | **14.44*** | 1100 | 73.36 | 13.76 | 1000 |

Table 1: Comparison of the *K*-Center-based DIVERSEEVOL method with alternative sampling strategies and "strong" baselines using the full source data. Metrics include relative scores (*RS*), win-and-tie rate (*WTR*), and optimal data sizes ($N_{best}$) behind the peak *RS*. If the best *WTR* is obtained with fewer data than $N_{best}$, it is marked with *, otherwise #. The gray-shaded rows are models using the entire source datasets as strong benchmarks. The best results are in **bold**; the second-best is underlined. Our DIVERSEEVOL approach consistently delivers high-quality results, matching or surpassing the strong baselines, with substantially fewer training samples.

| Source Datasets | # Samples | Annotator/Engine |
|---|---|---|
| Databricks-Dolly | 15011 | human |
| SelfInstruct-Davinci | 52002 | Text-Davinci-003 |
| SelfInstruct-GPT4 | 52002 | GPT-4 |

Table 2: Source datasets used in our experiments.

- **Win-And-Tie Rate (WTR)** vs. ChatGPT: Measures the frequency at which the chat model outperforms (WIN) or matches (TIE) the performance of ChatGPT:

$$\mathrm{WTR} = \frac{\sum_{q \in \text{testset}} \mathbb{I}(J(A_q^{\text{model}}) \geq J(A_q^{\text{chatgpt}}))}{|\text{testset}|} \tag{4}$$

**Configurations.** All our experiments utilize LLaMA-7B (Touvron et al., 2023) as the foundation LLM ($M_{pretrain}$). Unless stated otherwise, all iterative data sampling begins with an initial pool $P_0$ of 100 random samples. It spans $T = 10$ iterations with a new data point budget $k = 100$. For instruction-tuning each chat model, we finetune the LLaMA model for 3 epochs with the batch size set to 128 and the learning rate set to $2 \times 10^{-5}$. The Alpaca-style template (Taori et al., 2023) is adopted to prepare input from the instruction data.

## 4.2 Main Results

Utilizing our DIVERSEEVOL approach, chat models evolve in their instruction-following capability as the training data pool progressively augments through our *K*-Center-Sampling strategy.

Tab. 1 compares our *K*-Center-based DIVERSEEVOL method with alternative sampling strategies and strong baselines trained on full source data (`*Full Data`). The metrics reported include Relative Scores (*RS*), Win-and-Tie Rates (*WTR*), and the optimal data sizes ($N_{best}$) associated with peak *RS*. With the *K*-Center-based DIVERSEEVOL strategy, our chat models frequently match or exceed the performance of the strong baselines with far fewer training samples. On the human-annotated source dataset *Databricks-Dolly-15K*, our method consistently achieves the best *RS* and *WTR* across benchmarks, surpassing the baseline finetuned on the entire 15K data by a considerable margin with merely 700 or 1100 samples, corresponding to less than 4% data size. On the *SelfInstruct-52K* data generated by *Text-Davinci-003* or *GPT4*, DIVERSEEVOL achieves similar effects of top performance surpassing the strong baselines on the majority of metrics using only 2% or less of the 52K source data ($\leq$ 1100 samples). Even

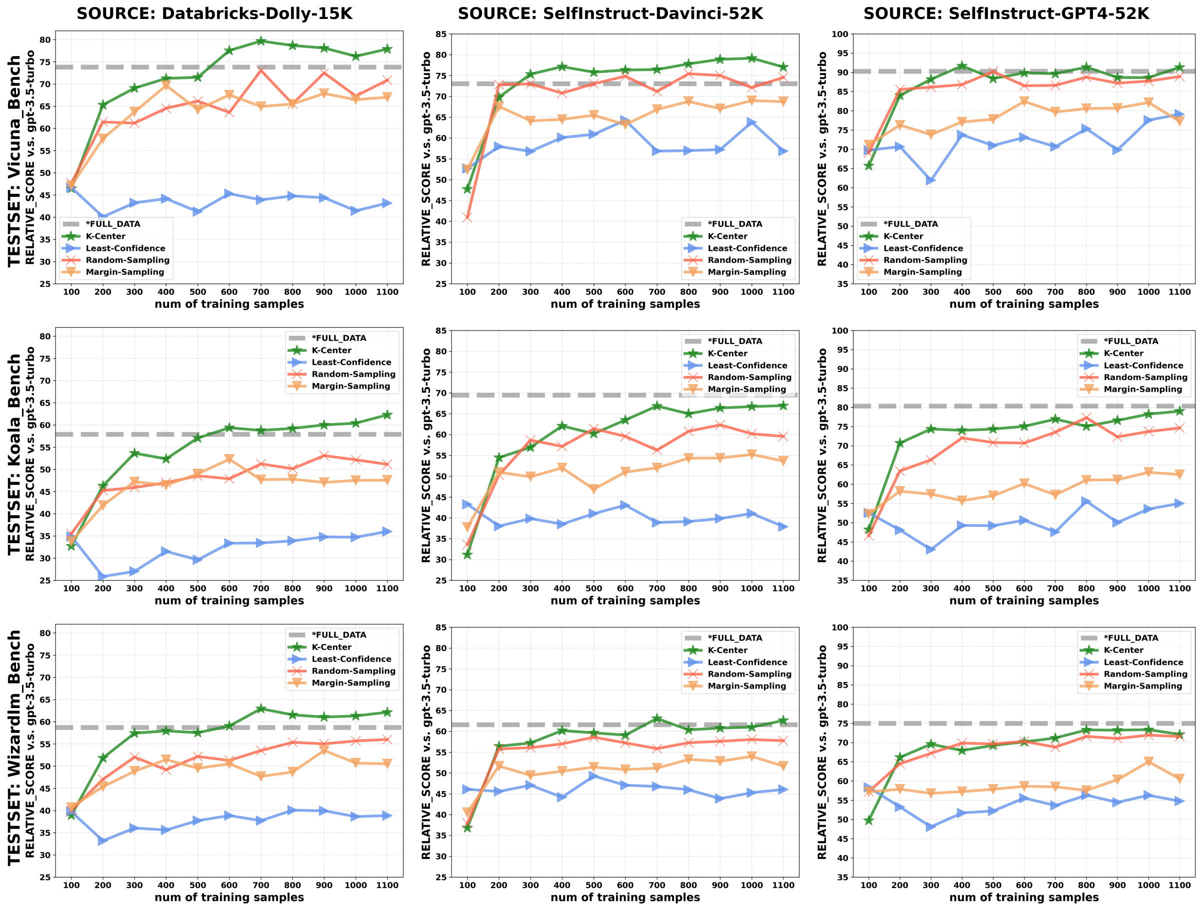


Figure 2: Performance evolution of chat models across various source datasets using our proposed $K$-Center based DIVERSEEVOL and alternative sampling approaches. The Y-axis represents relative scores (*RS*) with respect to ChatGPT, while the X-axis indicates the number of training samples. The curves demonstrate the rapid proficiency gains achieved by the DIVERSEEVOL approach, matching or often outpacing strong baselines (`*Full Data`) trained on the full dataset with only a significantly small fraction of the data.

on benchmarks where our method does not stand out as the best performer, it achieves at least the second-best results behind the strong baselines by a small margin, such as in the case of *RS* with the highest gap of mere 2.55 on Koala-Bench using the *SelfInstruct-Davinci* source data. This unambiguously shows the effectiveness and efficiency of our proposed DIVERSEEVOL data selection strategy. In contrast, other sampling strategies like random sampling or confidence-based selection (e.g., Least-Confidence, Margin-Sampling as discussed in §4.1) tend to underperform or at best only seldom match the strong baselines, which largely falls behind DIVERSEEVOL's overall performance.

Fig. 2 provides a complementary view to Tab. 1, illustrating the exact trajectory of performance evolution (measured by *RS*) with iteratively extended training data pool. The trend line in this figure is revealing. Our $K$-Center based DIVERSEEVOL models (marked in green) start to match or surpass the strong baselines trained on the complete dataset (`*Full Data`) remarkably quickly, namely in only a few iterative steps, requiring several hundred samples selected from the source dataset. On the source dataset *Databricks-Dolly-15K*, our method manages to match the upper bound-baseline with only 600 samples (4%) across test sets. Compared with alternative sampling strategies, our $K$-Center-based DIVERSEEVOL method also consistently stands out as the top-performing curve, showing better scores throughout the iteration, regardless of source datasets or testing benchmarks.

### 4.3 Analyses

We provide further analyses of the two main factors behind the effectiveness of DIVERSEEVOL, namely: diversity of selected datasets, and the dynamic iteration scheme.

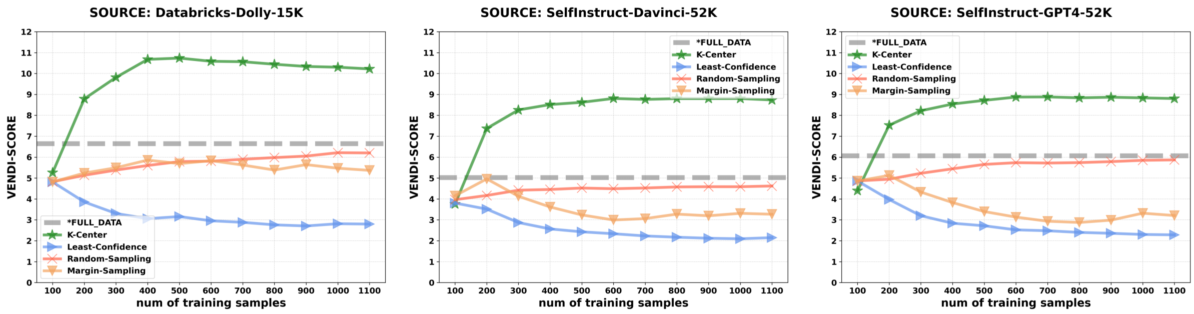


Figure 3: Diversity evolution in the selected training data pool from three source datasets. The Y-axis denotes the Vendi-Score for measuring diversity, and the X-axis shows increasing data size. The gray line (`*Full Data`) represents original source dataset diversity. The contrasting curves highlight our $K$-center approach's early and sustained enhancement of data diversity.

| $K$-Center | | Vicuna-Bench | | | Koala-Bench | | | Wizardlm-Bench | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $N$ | 300 | 700 | 1100 | 300 | 700 | 1100 | 300 | 700 | 1100 |
| **Iterative (DiverseEvol)** | $RS$ | **69.09** | **79.69** | **77.90** | **53.65** | **58.78** | **62.29** | **57.42** | **62.94** | **62.15** |
| **One-Time Direct Sampling** | $RS$ | 67.38 | 73.90 | 73.21 | 51.42 | 58.10 | 57.56 | 50.94 | 61.82 | 60.97 |

Table 3: Comparison of performance between the dynamic, iterative sampling scheme as in DiverseEvol and one-time data selection method of directly sampling to a given data size. With the same $K$-Center selection algorithm, this table shows that the iterative approach consistently outperforms the method of direct sampling for once across different data volumes, highlighting the importance of iterative feedback in improving chat model capabilities.

**Diversity.** Based on the main results reported in Tab. 1 and Fig. 2, we believe that maintaining high diversity in the training data pool is crucial for a successful instruction-tuning dataset. This is also exactly the design principle behind our $K$-Center based DiverseEvol that seeks to find the most representative subset of a source data pool, constituting the most diverse cover of the source dataset (§3.2). Given that diversity is a focal point in our method, we also explicitly assess data diversity using an automatic metric, **Vendi-Score** (Friedman and Dieng, 2022) that measures the datapoint distribution's diversity based on their embeddings' similarity matrix. To testify to the pivotal role of diversity, we thus conduct empirical analyses from the following two angles.

First, we use the above diversity metric to quantitatively measure the level of data diversity achieved by our $K$-Center-based method, compared to the original dataset diversity and other sampling methods. In Fig. 3, we present the Vendi-Score of the maintained training data pool $P_t$ at each iteration step $t$, in line with the X-axis in Fig. 2. As shown in the figure, our $K$-Center data selection algorithm (Alg. 1) significantly boosts the diversity of the training data pool at an early stage, surpassing the diversity of the original source dataset and all other sampling methods. This demonstrates the effectiveness of our $K$-center-based sampling in selecting datapoints that constitute the most diverse cover of the source dataset.

Second, to further demonstrate the diversity of the training dataset as a key contributor to model performance, we directly control the Vendi-Score as a diversity variable and report how varying the level of diversity in the training dataset leads to varying instruction-tuned chat model performance. Using *Databricks-Dolly* as an example source dataset, we perform independent random sampling, devoid of any algorithmic influence, for multiple iterations to achieve specific Vendi-Scores for predetermined training data sizes. Our experiment comprises three distinct training data volumes: 300, 700, 1100. For each volume, we target three levels of diversity, measured by Vendi-Score of ranges: $[3, 4]$, $[5, 6]$, and $[9, 10]$. A negligible deviation of $\pm 0.2$ is observed, because larger data sizes make it harder to mine more or less diverse samples given the randomness of the procedure. Subsequently, we train chat models using datasets behind the highest, median, and lowest range of Vendi-Score, representing high, medium, and low data diversity, respectively. In Fig. 4, we show the resulting chat model performance measured by Rel-

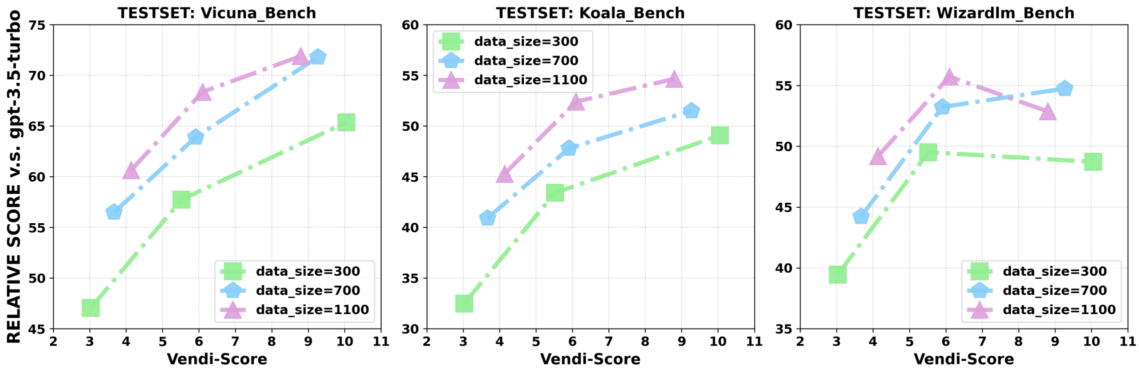


Figure 4: Performance of instruction-tuned chat models in relation to Vendi-Score of their training datasets, illustrating the influence of data diversity. The three distinct curves correspond to training data volumes of 300, 700, and 1100. A consistent trend of performance enhancement is observed with increased dataset diversity across most benchmarks, with only minor deviations seen on the Wizardlm-Bench.

ative Score (RS) v.s. ChatGPT in regard to Vendi-Score of its training dataset, signifying the level of diversity. Each curve represents a controlled total training data size. Evidently, the degree of diversity in the training data pool significantly influences the resulting chat model's performance regardless of data volume. We observe an nearly consistent boost of chat model performance as we maintain a more diverse training data pool almost across testing benchmarks, except for marginal deviations on the Wizardlm-Bench. The sheer elevation of *RS* as a result of increased dataset diversity is striking, often reaching over 10 points, especially from the very lowest range of Vendi-Score to the medium level. This effectively proves data diversity as a key factor in boosting instruction-tuned chat model capability.

**Dynamic Iteration.** Another distinguishing aspect of our methodology is its iterative nature in data selection, which we demonstrate is crucial in bolstering the chat model's ability to follow instructions. Using the *Databricks-Dolly* source dataset as an example case, we contrast our primary iterative approach, where the chat model's data pool incrementally expands, against an alternative strategy where data is directly sampled at three different volumes: 300, 700, and 1100. Both methods employ the same $K$-Center selection method, with the initial 100 samples chosen randomly.

Tab. 3 vividly demonstrates the differences in performance. Regardless of the final training data size, our proposed iterative approach (DIVERSEEVOL), mirroring the results in Tab. 1 with corresponding $N_{best} = N$, consistently outperforms the method of directly sampling the same data volume (One-Time Sampling). Notably, while the $K$-Center sampling technique remains identical across both approaches, the obvious performance variance underscores the pivotal role of iterative feedback. Such signals, derived from the trained chat model at every iterative step, guides subsequent data selections and establishes a progressive learning mechanism that capitalizes on insights from prior iterations. This contrasts sharply with direct sampling, which misses out on leveraging the experience accrued from past models, leading to suboptimal results. Therefore, our approach enables models to truly "evolve" itself over iterations, using insights from previous stages to inform future training data selection. This iterative feedback loop starkly outperforms a one-off decision-making process, underlining its essential role in enhancing model performance.

## 5 Conclusion

We introduced DIVERSEEVOL, a self-evolving method for efficient instruction tuning of LLMs. Relying on an iterative scheme, DIVERSEEVOL progressively improves itself by selecting diverse subsets from vast instruction data using the $K$-Center strategy without seeking any external supervision. Empirical results affirm that, with less than 4% of the original data size, our method matches or surpasses strong baselines in performance. Future endeavors can delve into leveraging our method on larger instruction datasets for potentially even more refined results. Building upon the foundation laid by DIVERSEEVOL, more advanced algorithms of diverse sampling also promise to enhance model performance further.

## Limitations

The $K$-Center sampling method in DIVERSEEVOL involves computing distances between high-dimensional embeddings of datapoints. If the source dataset further increases in size, this computation may impose a considerable expense on the GPU memory. Furthermore, our evaluation outcomes rely heavily on GPT4-judge. Despite our attempts to obtain a more deterministic result by setting the querying temperature to 0, and to address position-bias through two-time querying with model responses in alternating positions, the evaluation process may still be influenced by inherent biases within the GPT4 model.

## References


Waseem AlShikh, Manhal Daaboul, Kirk Goddard, Brock Imel, Kiran Kamble, Parikshith Kulkarni, and Melisa Russak. 2023. Becoming self-instruct: introducing early stopping criteria for minimal instruct tuning. *arXiv preprint arXiv:2307.03692*.

Yihan Cao, Yanbin Kang, and Lichao Sun. 2023. Instruction mining: High-quality instruction data selection for large language models. *arXiv preprint arXiv:2307.06290*.

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

Wei-Lin Chiang, 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.

Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. 2021. Batch active learning at scale. *Advances in Neural Information Processing Systems*, 34:11933–11944.

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.

Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *AAAI*, volume 5, pages 746–751.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*.

Dan Friedman and Adji Bousso Dieng. 2022. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*.

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

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. 2023. Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Luke Zettlemoyer, Omer Levy, Jason Weston, and Mike Lewis. 2023. Self-alignment with instruction backtranslation. *arXiv preprint arXiv:2308.06259*.

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. *arXiv e-prints*, pages arXiv–2308.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. 2011. Reading digits in natural images with unsupervised feature learning.

OpenAI. 2023. Gpt-4 technical report.

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

Ozan Sener and Silvio Savarese. 2017. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*.

Claude Elwood Shannon. 2001. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55.

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`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Dan Wang and Yi Shang. 2014. A new active labeling method for deep learning. In *2014 International joint conference on neural networks (IJCNN)*, pages 112–119. IEEE.

Kai Wei, Yuzong Liu, Katrin Kirchhoff, and Jeff Bilmes. 2013. Using document summarization techniques for speech data subset selection. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 721–726.

Gert W Wolf. 2011. Facility location: concepts, models, algorithms and case studies. series: Contributions to management science. *International Journal of Geographical Information Science*, 25(2):331–333.

Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. 2023a. Expertprompting: Instructing large language models to be distinguished experts. *arXiv preprint arXiv:2305.14688*.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment.

## A GPT4-Judge Template

We conduct automatic evaluation of chat model's performance using GPT4 as judge (§4.1). Given a question (i.e., instruction) from test set and answers generated by two models, here's the template we used, adapted from (Chiang et al., 2023):

**Template for GPT4-Judge**

[Question]
`{instruction}`

[The Start of Assistant 1's Answer]
`{answer-of-chatbot1}`
[The End of Assistant 1's Answer]

[The Start of Assistant 2's Answer]
`{answer-of-chatbot2}`
[The End of Assistant 2's Answer]

[System]
We would like to request your feedback on the performance of two AI assistants in response to the user question displayed above. Please rate the helpfulness, relevance, accuracy, level of details of their responses. Each assistant receives an overall score on a scale of 1 to 10, where a higher score indicates better overall performance. Please first output a single line containing only two values indicating the scores for Assistant 1 and 2, respectively. The two scores are separated by a space. In the subsequent line, please provide a comprehensive explanation of your evaluation, avoiding any potential bias and ensuring that the order in which the responses were presented does not affect your judgment.

Throughout our experiments, the specific model versions of our OpenAI's API calls are: *GPT-3.5-TURBO-0613* and *GPT-4-0613*.