InsBank: Evolving Instruction Subset for Ongoing Alignment

Jiayi Shi[1†], Yiwei Li[1†], Shaoxiong Feng[2], Peiwen Yuan[1], Xinglin Wang[1], Yueqi Zhang[1], Chuyi Tan[1], Boyuan Pan[2‡], Huan Ren[2], Yao Hu [2], Kan Li[1‡]

[1] School of Computer Science, Beijing Institute of Technology
[2] Xiaohongshu Inc
{shijiayi,liyiwei,peiwenyuan,wangxinglin}@bit.edu.cn
{shaoxiongfeng2023}@gmail.com {panboyuan,qiaoen,xiahou}@xiaohongshu.com
{zhangyq,tanchuyi,likan}@bit.edu.cn

Abstract

Large language models (LLMs) typically undergo instruction tuning to enhance alignment. Recent studies emphasize that quality and diversity of instruction data are more crucial than quantity, highlighting the need to select diverse, high-quality subsets to reduce training costs. However, how to evolve these selected subsets alongside the development of new instruction data remains insufficiently explored. To achieve LLMs' ongoing alignment, we introduce Instruction Bank (**InsBank**), a continuously updated repository that integrates the latest valuable instruction data. We further propose Progressive Instruction Bank Evolution (**PIBE**), a novel framework designed to evolve InsBank effectively and efficiently over time. PIBE employs a gradual data selection strategy to maintain long-term efficiency, leveraging a representation-based diversity score to capture relationships between data points and retain historical information for comprehensive diversity evaluation. This also allows for flexible combination of diversity and quality scores during data selection and ranking. Extensive experiments demonstrate that PIBE significantly outperforms baselines in InsBank evolution and is able to extract budget-specific subsets, demonstrating its effectiveness and adaptability.[1]

1 Introduction

Instruction fine-tuning is widely adopted to refine pre-trained LLMs to accurately understand human instructions and provide precise, pertinent and harmless responses (Longpre et al., 2023; Qin et al., 2024a). LIMA (Zhou et al., 2023a) has proved that the quality and diversity of instruction data are significantly more critical than its sheer quantity for training, motivating recent efforts in instruction data selection to reduce unnecessary training costs by eliminating low-quality and redundant data (Qin et al., 2024a). However, how to evolve the selected instruction subset in parallel with the development of the instruction data remains underexplored.

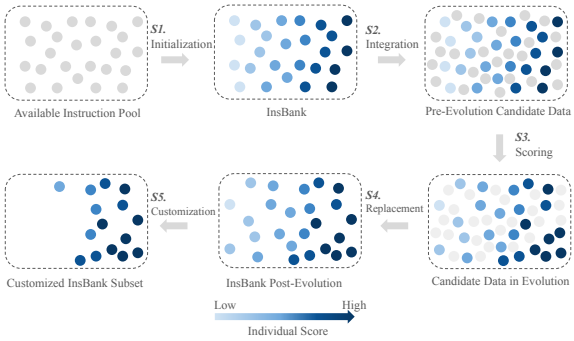

Figure 1: Illustration of InsBank evolution. It is initialized by data selection on all current available instruction data, and it will evolve itself as long as new instruction data are proposed. A smaller training subset can be obtained from InsBank according to user training budget.

Specifically, with the continuous emergence of instruction datasets (The timeline of part instruction datasets is shown in Appendix A), it becomes necessary to regularly update the instruction subset to incorporate the latest advanced instruction data in order to ensure ongoing improvements in the alignment capabilities of LLMs. Simultaneously, the subset size must be controlled to avoid excessive growth that could lead to increased training costs. To address these practical challenges, we propose a novel concept termed **InsBank** (Instruction Bank). InsBank is designed to support instruction subset evolution with two key properties: (1) To prevent unbounded growth, InsBank maintains a constant size by replacing low-quality old samples with an equal number of high-scoring new ones during evolution. (2) Samples in InsBank are ranked according to their overall scores to enable users to extract subsets that are tailored

[†]Equal contributions.
[‡]Corresponding authors.
[1]Our code has been released on https://github.com/jiayinlp/InsBank

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 220–238
November 4-9, 2025

to specific training budgets, simply by selecting the top-ranked samples. The evolution process of InsBank is illustrated in Figure 1.

As the scale of existing instruction sets continues to grow(Qin et al., 2024a; Longpre et al., 2023; Wang et al., 2023; Xu et al., 2023)—reaching millions or even billions of instances—the cost of exhaustively traversing all candidate data during each InsBank evolution becomes prohibitively high. To address this challenge, we propose Progressive Instruction Bank Evolution (**PIBE**), a method designed for continuous and efficient selection of the optimal instruction subset. PIBE evolves InsBank in a gradual manner, ensuring long-term efficiency. Unlike the naive approach, it significantly reduces the cost of evolution by excluding previously filtered-out data and focusing solely on newly proposed samples and the current InsBank.

Additionally, the orderliness of InsBank calls for an overall score that integrates both individual quality and diversity signals. While quality scores can be readily obtained through manual or model-based annotation, measuring individual diversity requires global comparisons among candidates. Unfortunately, existing instruction data selection methods struggle to effectively represent and combine quality and diversity for ranking purposes. This challenge is further exacerbated by the absence of historical data, which alters the distribution of candidates and underscores the need to retain historical data distributional information during evolution. Existing diversity-driven data selection methods (Liu et al., 2024; Wu et al., 2023) typically fall into two categories: k-nearest neighbor (k-NN) approaches (Dong et al., 2011) and geometry-based coreset sampling methods (Guo et al., 2022). Both of them rely exclusively on local information from a limited number of neighboring points, which limits their ability to capture global relationships and provide reliable individual diversity scores for ranking. Furthermore, they lack mechanisms to preserve information about previously discarded data, making them ill-suited for progressive selection. Inspired by Affinity Propagation (Frey and Dueck, 2007), we frame InsBank data selection as an exemplar election process, where the representativeness of each data point is quantified through an iterative voting mechanism. The representativeness further serves as the individual diversity score, and the voting results are passed to the next iteration as historical information to preserve the distribution of absent data. Moreover, existing data selection methods either prioritize quality or diversity (Chen et al., 2024), or address them sequentially (Liu et al., 2024), failing to consider both aspects equally. Conversely, our diversity score integrates seamlessly with the quality score, enabling comprehensive and flexible instruction selection and InsBank ranking.

We simulate the instruction set development with five datasets and perform InsBank evolution on them with PIBE and we elaborate on the rationale for selecting these datasets in Appendix J.1. We evaluate the general instruction following capability of fine-tuned models on AlpacaEval (Li et al., 2023b), MT-Bench (Zheng et al., 2023), IFEval (Zhou et al., 2023b), OpenLLM Leaderboard (Beeching et al., 2023) and FollowBench (Jiang et al., 2024). Experimental results show that PIBE outperforms the baselines and successfully evolves the instruction bank in parallel with the development of instruction sets. Besides, analysis on orderliness of InsBank indicates that users can flexibly select a smaller subset based on their budget. Ours contributions can be summarized as follows:

- We propose InsBank, a dynamic framework for evolving instruction subsets alongside the development of instruction data, enabling continuous alignment improvements.
- We develop Progressive Instruction Bank Evolution (PIBE), an efficient approach that leverages a memory-enhanced diversity score and seamlessly integrates it with quality scores for optimal subset selection.
- We introduce a unified scoring system for individual samples, ensuring an ordered InsBank and enabling flexible extraction of high-quality subsets tailored to user budgets.
- Extensive experiments demonstrate that PIBE not only outperforms baseline methods in evolving InsBank but also provides flexible, budget-aware data selection, highlighting its effectiveness and adaptability.

2 Preliminaries

2.1 Instruction Data Selection Problem

Following Liu et al. (2024), given a collection of instruction data $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ where x_i is an individual instruction-response pair, data selection selects an instruction subset $\mathcal{P}_\pi^m$ of size m from $\mathcal{X}$, where π is the data selection strategy. Denote the performance evaluation function for π as Q, the optimal data selection strategy π^* with subset size

m satisfies:

$$\pi^* = \arg\max_{\pi} Q(\mathcal{P}_\pi^m) \tag{1}$$

2.2 Selection Metrics

Previous research (Liu et al., 2024; Qin et al., 2024a) highlight that the effectiveness of instruction set selection depends on both quality and diversity. In line with this, we focus on the two aspects in this paper:

Quality of instruction data primarily refers to the accuracy and rationality which estimate the consistency and coherence of the instruction context, as well as whether the response accurately corresponds to the instructions (Qin et al., 2024a). In this work, we adopt the quality evaluation model of DEITA (Liu et al., 2024) for quality annotation.

Diversity of instruction data is critical to the generalization ability of the trained model (Qin et al., 2024a). There are currently two major approaches to measure diversity: k-nearest neighbor (k-NN) (Dong et al., 2011) and geometry-based coreset sampling (Guo et al., 2022). The kNN approach measures sample's diversity by its distance to its j-th k-nearest neighbor (k-NN) with the help of text embeddings as shown in Eq. 2:

$$kNN_i^j = d(e(x_i), e(N_j(x_i))) \tag{2}$$

where $N_j(x_i)$ denotes the j-th closest neighbor of x_i in the embedding space projected by $e(\cdot)$, and $d(\cdot,\cdot)$ calculates the distance between x_i and $N_j(x_i)$. The geometry-based coreset sampling approach is to find the most informative-and-diverse subset that represents the entire dataset the most through controlling the minimum distance between any two samples for subset selection (Guo et al., 2022; Sener and Savarese, 2018). However, both methods rely solely on local information from nearby points, making it difficult to capture the global distribution relationships or utilize historically eliminated points, resulting in inadequate individual diversity scores for subset evaluation.

2.3 Affinity Propagation

Affinity Propagation (AP) (Frey and Dueck, 2007) is a clustering algorithm that leverages message-passing to uncover the global distribution of data. It identifies exemplars by iteratively transmitting two kinds of messages between data points:

- **Responsibility** ($R[i,k]$) This message sent from point i to point k represents how suitable point k is to serve as the exemplar for point i.
- **Availability** ($A[i,k]$) This message sent from point k to point i represents how appropriate it would be for point i to choose point k as its exemplar, taking into account the current responsibilities sent from other points to k.

The messages are updated iteratively based on the rules as shown in Eq. 3. Here, $S[i,k]$ represents the similarity between point i and point k where $i \neq k$. And $S[k,k]$ is filled by the predefined preference value which represents the preference for sample i as an exemplar.

$$\begin{aligned}
&R[i,k] \leftarrow S[i,k] - \max_{k' \neq k} \left\{ A[i,k'] + S[i,k'] \right\}, \\
&A[i,k] \leftarrow \min \left\{ 0, R[k,k] + \sum_{i' \notin \{i,k\}} \max\left\{0, R[i',k]\right\} \right\}, \\
&A[k,k] \leftarrow \sum_{i' \neq k} \max\{0, R[i',k]\},
\end{aligned} \tag{3}$$

At any given moment, the clustering result can be determined by summing R and A. For x_i, let k' be the index that maximizes $A[i,k] + R[i,k]$, the conclusion are as follows: (1) if $i = k'$, then x_i is a cluster center, (2) if $i \neq k'$, then x_i belongs to the cluster center $x_{k'}$. That is, for $R + A$, the i-th row represents the votes cast by x_i for different points to represent itself, while the j-th column represents the votes received by x_j. Based on this, we obtain the representativeness of x_i according to the voting results by subtracting the votes cast by x_i for other samples from the votes received by x_i. This result serves as individual diversity score.

3 Progressive Instruction Bank Evolution

In this section, we provide a detailed explanation of PIBE, whose pipeline is depicted in Figure 2.

3.1 Gradual Evolution Formulation

In this work, we propose the instruction subset evolution task to build the InsBank. Denoting current available instruction data as $\mathcal{X}_0$, the instruction bank $\mathcal{B}_\pi^{0,m}$ of size m is initialized through data selection which can be presented as $\mathcal{B}_\pi^{0,m} = \pi(\mathcal{X}_0)$. Then, when new instruction dataset $\mathcal{X}_1$ is proposed, $\mathcal{B}_\pi^{0,m}$ should evolve itself to adapt to changes in data distribution. The naive manner of InsBank evolution can be represented as $\mathcal{B}_\pi^{1,m} = \pi(\mathcal{X}_0, \mathcal{X}_1)$ which can be extended to $\mathcal{B}_\pi^{t+1,m} = \pi(\mathcal{X}_0, ..., \mathcal{X}_t, \mathcal{X}_{t+1})$ for future evolution. However, this manner requires substantial storage and computational resources to calculate

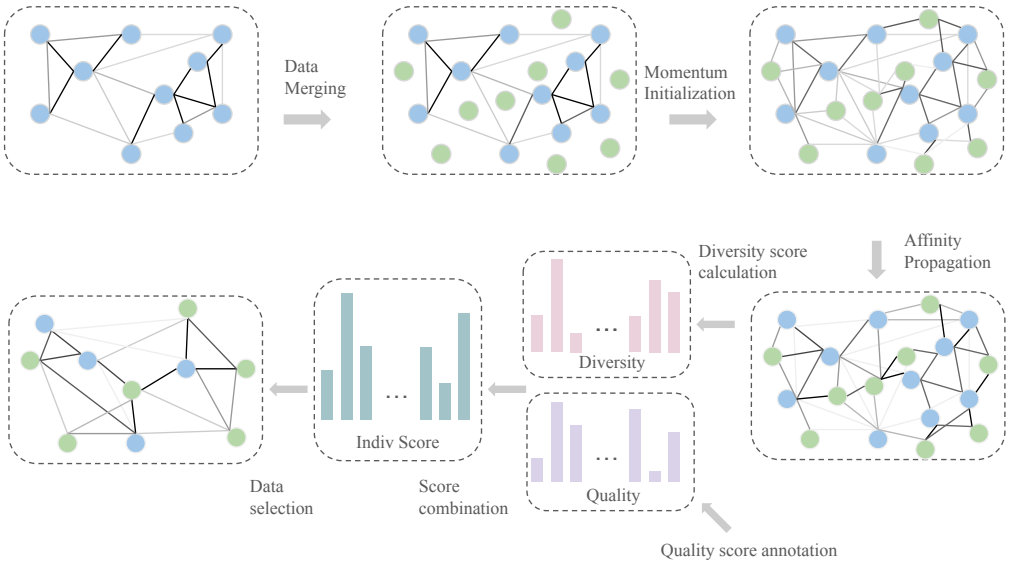

Figure 2: The framework of PIBE begins by integrating newly proposed candidates with the existing InsBank data and initializing momentum information based on historical records. Then, affinity propagation incorporating the momentum is applied to compute diversity scores. Subsequently, the quality scores obtained via model-based annotation are combined with the diversity scores to produce an individual overall score. Finally, the top-k samples with the highest overall scores are selected to form the evolved InsBank where k is the budget.

diversity scores as t continues to increase. To improve the long-term evolution efficiency, we propose a gradual manner where only the newly proposed instruction data $\mathcal{X}_{t+1}$ along with the data participated in last round of evolution $\mathcal{X}_t + \mathcal{B}_\pi^{t-1,m}$ are involved into the current round of evolution, and the evolution can be represented as $\mathcal{B}_\pi^{t+1,m} = \pi(\mathcal{X}_{t+1}, \mathcal{X}_t + \mathcal{B}_\pi^{t-1,m})$.

In addition to the update of InsBank, we evaluate the diversity and quality of each sample x_i and provide an overall individual score for data ranking. Users can quickly select a smaller subset according to the data ranking to suit their own training budget.

3.2 Historical Information Flowing

Although a large amount of data is eliminated during InsBank evolution for efficiency, preserving their distribution information is crucial for maintaining InsBank's global representativeness. To address this, we introduce a momentum matrix based on historical voting results to retain the distribution information of excluded data, which flows across iterations, allowing filtered-out data to re-engage in future exemplar selection and preventing suboptimal global representativeness.

As described in Section 2.3, we evaluate individual diversity through AP. By analyzing the similarity between previously selected data and newly proposed candidates, we estimate the suitability of new data as exemplars for the existing data and vice versa, represented by the responsibility matrix.

Formally, let $\mathcal{X}'_t = \mathcal{X}_t \cup \mathcal{B}_\pi^{t-1,m}$ denote the full candidate data set from the previous round of InsBank evolution, and $\mathcal{X}'_{t+1} = \mathcal{X}_{t+1} \cup \mathcal{B}_\pi^{t,m}$ denote the full candidate data set of the $(t+1)$-th evolution round. Then, the matrix Sim_{t+1} of size $|\mathcal{X}'_t| \times |\mathcal{X}_{t+1}|$ represents the cosine similarity between $\mathcal{X}'_t$ and $\mathcal{X}_{t+1}$. Given the historical information matrix H_t of size $|\mathcal{X}'_t| \times |\mathcal{X}'_t|$, representing the responsibility matrix stored from the t-th round of InsBank evolution, we derive the momentum responsibility matrix M_t using H_t and Sim_{t+1}:

$$w_{jk} = \frac{Sim[j,k]}{\sum_{l=1}^{|X'_t|} Sim[l,k]}, \qquad M_t[i,k] = \sum_{j=1}^{|X'_t|} w_{jk} * R_t[i,j] \tag{4}$$

$$M_t[i,k] = \sum_{j=1}^{|X'_t|} w_{ij} * R_t[j,k] \tag{5}$$

This allows the filtered-out data to participate in exemplar election during future history-aware AP processes.

The structure of M_t is depicted in Appendix B. The top-left part of M_t contains responsibility values between data in $\mathcal{B}_\pi^{t,m}$, taken directly from

H_t. The top-right part represents the suitability of newly proposed candidate data as exemplars for previously selected data, estimated using Eq. 4. Similarly, the bottom-left part represents the suitability of previously selected data as exemplars for newly proposed candidate data, estimated using Eq. 5. The bottom-right section is filled with the median values of the other three sections.

We regard M_t as a continuously decaying momentum term for historical information preserving. Specifically, we first calculate R^i_{t+1} by Eq. 3. Then, we apply a weighted sum of M_t and R^i_{t+1} to recall the historical information as shown in Eq. 6,

$$R^i_{t+1} = \alpha_i \cdot M_t + (1-\alpha_i) \cdot (\beta \cdot R^i_{t+1} + (1-\beta) \cdot R^{i-1}_{t+1}) \quad (6)$$

where $\alpha_i = \lambda \cdot \alpha_{i-1}$ is the momentum coefficient with a decay rate of λ, and β is the official AP damping rate (Frey and Dueck, 2007). Finally, A^i_{t+1} is calculated by Eq. 3. All α, λ and β are predefined hyperparameters.

3.3 Representativeness Scoring

The individual representativeness score encapsulates the results of the exemplar election, reflecting both how willing other samples are to be represented by a specific sample and how unwilling the specific sample is to be represented by others. As explained earlier, the responsibility value $R[i,k]$ indicates the suitability of x_k to serve as the exemplar for x_i, while the availability value $A[i,k]$ reflects the appropriateness of x_i selecting x_k as its exemplar. The combined value $(A+R)[i,k]$ represents the total evidence supporting x_i's selection of x_k as its exemplar (Frey and Dueck, 2007). Thus, the sum of the k-th column of $A+R$ can be interpreted as the total votes received by x_k, and the sum of the i-th row of $A+R$ represents the total votes cast by x_i for different samples. Defining $Z = A + R$, the representativeness score of x_k is then computed using Eq. 7.

$$s^k_{rep} = \sum_{i=1}^{|X'_{t+1}|} Z[i,k] - \sum_{i=1}^{|X'_{t+1}|} Z[k,i] + Z[k,k] \quad (7)$$

3.4 Integration of Diversity and Quality

Both data quality and data diversity are crucial for instruction tuning, yet existing methods often focus on one or address them sequentially. We combine quality and diversity scores in three ways, both preceded by min-max normalization (Eq. 8) to ensure scale consistency, where s^k_q refers to the quality score of x_k, and s^k_{rep} refers to the corresponding diversity score.

$$\begin{aligned} s'^k_{rep} &= \frac{s^k_{rep} - \min\limits_{x_i \in B^m_t} s^i_{rep}}{\max\limits_{x_i \in X'_{t+1}} s^i_{rep} - \min\limits_{x_i \in B^m_t} s^i_{rep}}, \\ s'^k_q &= \frac{s^k_q - \min\limits_{x_i \in X'_{t+1}} s^i_q}{\max\limits_{x_i \in X'_{t+1}} s^i_q - \min\limits_{x_i \in X'_{t+1}} s^i_q} \end{aligned} \quad (8)$$

$$s^k = s'^k_{rep} + \gamma \cdot s'^k_q. \quad (9)$$

$$s^k = (1 + s'^k_{rep}) * (1 + s'^k_q)^\gamma \quad (10)$$

Eq. 9 and Eq. 10 illustrate the calculation of the individual overall score using the additive and multiplicative approaches, respectively, where γ is the weighting coefficient that controls the focus between diversity and quality.

In practice, we observe that further improving quality beyond a certain level can reduce the fine-tuned model's performance. Additionally, when combining quality and diversity using linear methods, diversity scores often dominate the selection process. This occurs because quality, as a linear score, increases at a constant rate, even when excessively large values provide diminishing benefits. More details can be found in our experimental analysis of score combination (Section 4.4).

To address this, we design a nonlinear mapping function for quality scores, shown in Eq. 11. Here, Q_p denotes the p-th percentile, r_l and r_h represent the lower and upper percentiles, S'_q refers to the scaled quality scores, and $\sigma(\cdot)$ is the sigmoid function. The function, illustrated in Figure 7, leverages the sigmoid's steepness in $(-2, 2)$ to enhance the distinguishability of scores within $[\tau_l, \tau_h]$, while flattening growth for scores above τ_h. Data below τ_l are less considered, as such low-quality data are rarely selected into InsBank. Finally, we combine diversity with the nonlinear-mapped quality scores.

$$\begin{aligned} \tau_l &= Q_{r_l}(S'_q) \\ \tau_h &= Q_{r_h}(S'_q) \\ c_{mul} &= 4/(\tau_h - \tau_l) \\ c_{sub} &= \tau_l + 2/c_{mul} \\ s''^k_q &= \sigma((s'^k_q - c_{sub}) * c_{mul}) \end{aligned} \quad (11)$$

After getting the overall scores, in addition to serving as the criterion for InsBank data selection, users can quickly select a smaller subset according to the data ranking to suit their own training budget.

Method	Llama3-8B			Qwen2.5-7B			Mistral-7B		
	AlpacaEval	MT-Bench	IFEval	AlpacaEval	MT-Bench	IFEval	AlpacaEval	MT-Bench	IFEval
Full	19.07	5.88	40.29	20.37	6.11	41.37	13.12	4.98	**35.25**
Random	17.93	5.13	38.13	22.80	6.00	43.53	11.93	4.39	9.95
kCenter	15.28	4.99	37.29	27.39	6.12	46.40	9.20	3.97	1.92
DEITA	43.60	6.03	38.25	50.43	6.86	45.44	28.82	4.93	33.57
kNN_1	40.62	6.04	38.49	46.96	6.62	45.56	26.62	4.91	33.81
PIBE (ours)	**44.84**	**6.23**	**40.89**	**51.55**	**6.88**	**46.76**	**29.48**	**5.03**	29.38

Table 1: Comparison between different methods. For AlpacaEval and MT-Bench, we employ gpt-4o as annotator. The **bold** text indicates the best results, and the underlined text represents the second-best results. The results of more base models can be found in Appendix J.5.

4 Experiment

4.1 Experimental Setup

Candidate Instruction Data We aggregate five instruction datasets for general instruction following capability: Self-Instruct (Wang et al., 2023), Alpaca (GPT-4) (Peng et al., 2023), Dolly (Conover et al., 2023), ShareGPT[2] (Chiang et al., 2023) and WizardLM (alpaca) (Xu et al., 2023), resulting in a mixed dataset of 278k samples. The statistics of each dataset is presented in Table 6.

Training and Evaluation In this work, we fine-tune Llama3.2-1B, Llama3.2-3B, Llama3-8B (AI@Meta, 2024), Qwen2.5-7B, Qwen2.5-14B (Qwen Team, 2024) and Mistral-7B (Jiang et al., 2023) on the selected InsBank. Following DEITA (Liu et al., 2024), we set the size of InsBank to 6k for the convenience of subset evolution. We also experiment with InsBank size of 1k and 3k, and the results can be found in Appendix J.4. During training, we further restrict the trainable tokens and the number of conversation turns. We adopt AlpacaEval (Li et al., 2023b), MT-Bench (Zheng et al., 2023) and IFEval (Zhou et al., 2023b) for automatic model alignment performance evaluation. More details about training and evaluation can be found in Appendix C.

Baselines We compare proposed PIBE with the following baselines:

- **Full** Train model on all candidate data.
- **Random** Randomly select m samples from all candidate data.
- **kNN$_1$** Measure the diversity of one sample by its euclidean distance to the nearest neighbor (Eq. 2). The diversity score is first normalized and then combine with the normalized quality score by $s_i = (1 + kNN_1^i) * (1 + s_q'^i)^\gamma$ for data selection.
- **kCenter Greedy** (Sener and Savarese, 2018) The original kCenter Greedy algorithm is shown in Alg. 1. We take $\min_{x_j \in S_b} d(e(x_i), e(x_j))$ as the individual diversity score and combine it with quality score in the same manner of kNN$_1$.
- **DEITA** Traverse the instruction pool in descending order of quality scores and add a sample to the selected subset if its maximum cosine similarity with existing selected samples is below a threshold (Liu et al., 2024).

[2]We filter out incomplete conversations.

4.2 Performance of SFT with InsBank

Table 1 compares the performance of LLM trained on subsets selected by different approaches. PIBE consistently outperforms the baselines on such benchmarks, showing the superiority of our data selection method. We further fine-tune Qwen2.5 7B (Qwen Team, 2024) and Mistral 7B (Jiang et al., 2023) for robustness analysis, and the results exhibit the same trends, demonstrating that our method is effective across different models. We also report the quality and diversity of subsets selected by different methods in Table 2. From the results of data selection, PIBE and DEITA demonstrate higher quality and diversity compared to kCenter and kNN. DEITA produces subsets with the highest quality, primarily because it prioritizes quality during the data selection process by traversing candidates in descending order of quality. In contrast, PIBE treats quality and diversity equally, enabling the subset to achieve the highest diversity while maintaining decent quality. From the perspective of downstream task performance, models fine-tuned with high-quality data (DEITA, PIBE) generally outperform those fine-tuned on low-quality data (kCenter, kNN). However, despite achieving the highest quality, DEITA's downstream performance falls short of the more diverse PIBE, validating the importance of data diversity when the

quality level is acceptable.

4.3 Orderliness of InsBank

Each sample in the InsBank selected by PIBE is provided with an overall individual score reflects both the diversity and quality which shows the priority of each sample to be used to fine-tune models. We sort the InsBank in descending order based on the overall individual score, and compare the performance of models fine-tuned with the "top2k, mid2k, bottom2k" samples in InsBank. Here, we use the instruction subset obtained from the final evolution round, and restrict the trainable tokens to 0.9M and turns to 2.3k. The results are illustrated in Fig 3, showing that the top-ranked data generally achieved better performance, proving the orderliness of InsBank.

Metric	kCenter	DEITA	kNN_1	PIBE
Quality	4.37	**5.19**	4.82	5.13
Diversity	62.26	86.94	77.24	**91.84**

Table 2: The quality and diversity of subsets selected by different methods. The diversity here is measured by euclidean distance between data.

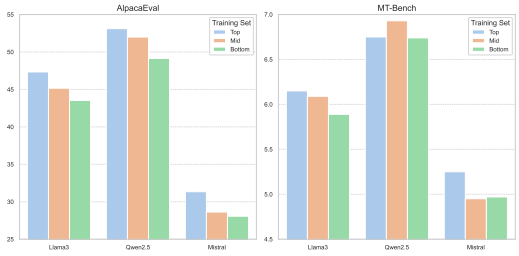

Figure 3: Results of orderliness experiment.

4.4 Analysis

In this section, we analyze the effectiveness of diversity and quality. We also experiment PIBE with different score combination methods. More analysis about overlap between progressive evolving and full data selection, InsBank evolution, PIBE hyperparameters, time costs and selected data quality distribution can be found in Appendix J.

Effectiveness of Diversity and Quality To validate the role of diversity in instruction data selection, we first construct a quality-controlled subset where all data have quality scores within the range of 4.5 to 5.0 (details in Appendix E). Using PIBE, we compute individual diversity scores for the subset, sort the data in descending order, and select the top 6k samples as the most diverse subset and the bottom 6k as the least diverse subset. The distributions of the two subsets are shown in Fig. 6. Before fine-tuning, we restrict the total trainable tokens to 2M. Results in Table 3 indicate that, with comparable quality, models trained on more diverse data achieve better performance.

Method	Qua	Div	AlpacaEval	MT-Bench
Top	4.84	81.14	**27.70**	**5.52**
Bottom	4.86	68.55	27.33	5.43

Table 3: The results of quality-controlled diversity effectiveness experiment. Qua refers to the average quality score, and Div refers to the average diversity score.

When it comes to quality, the improvement from extremely low to high quality is clearly beneficial, as extremely low-quality subsets often contain noisy data, such as irrelevant or incomplete responses. However, *is continuously improving quality always effective in the data selection process?* To address this, we compare model performance fine-tuned on data selected by the following strategies in the final evolution iteration: (1) **Diversity Greedy**: selecting data with the highest diversity scores; (2) **Quality Greedy**: selecting data with the highest quality scores; and (3) **PIBE**. The results shown in Table 4 reveal a clear trade-off between diversity and quality. A purely greedy approach focusing on either aspect leads to sub-optimal outcomes, while a balanced consideration of both proves more effective. This finding aligns with the main experiment results and suggests the existence of a balance point between diversity and quality, which we further investigate through the analysis of score combination.

Method	Qua	Div	AlpacaEval	MT-Bench
DG	5.02	93.06	**41.93**	**6.09**
QG	5.20	83.70	40.86	5.86
PIBE	5.13	91.84	44.84	6.23

Table 4: Analysis of diversity and quality contribution. Here, DG refers to diversity greedy, and QG refers to quality greedy

Analysis of Score Combination We experiment with the different combination methods to explore the contribution of quality and diversity in PIBE.

We first explore the multiplication manner and the addition manner, and the results are reported in Table 5. Overall, regardless of whether addition or multiplication is used as the combination method,

Param	AlpacaEval	MT-Bench	SP-Qua	SP-Div	Diff
Multiplication					
$\gamma = 1$	44.84	6.23	0.36	0.74	0.38
$\gamma = 2$	**46.77**	**6.15**	0.51	0.70	0.19
$\gamma = 3$	42.98	6.17	0.54	0.67	0.13
Addition					
$\gamma = 1$	44.84	6.13	0.44	0.72	0.28
$\gamma = 2$	**47.08**	**6.10**	0.54	0.68	0.14
$\gamma = 3$	44.53	6.09	0.56	0.64	0.08
Nonlinear					
$r_h = 0.80$	44.41	5.98	0.58	0.72	0.14
$r_h = 0.90$	44.84	6.19	0.62	0.70	0.08
$r_h = 0.95$	**47.58**	**6.36**	0.63	0.69	0.06

Table 5: The results of different combination methods. SP- refers to Spearman value, Diff refers to the difference value between SP-Qua and SP-Div.

the results exhibit a distinct trend of initially increasing and then decreasing as the influence of quality grows (i.e., with the increase of the γ value). This finding supports the hypothesis that a balance point exists between diversity and quality.

We analyze the correlation between quality and selection flags, as well as diversity and selection flags, for the top 12k data sorted by overall score (details in Appendix D). As shown in Table 5, Spearman for diversity consistently surpass those for quality, indicating diversity's priority during selection. While increasing γ reduces the gap, this approach presents limitations: (1) Even at $\gamma = 3$, a notable gap remains between SP-Qua and SP-Div, particularly with the multiplication method; (2) Increasing γ further improves downstream performance initially but leads to declines afterward.

Examining the quality distribution of selected data (Figure 10), we observe that $\gamma = 1$ includes some low-quality data, while $\gamma = 3$ selects excessive high-quality data. As discussed in Section 3.4, this stems from quality's linear nature. To address this, we use a nonlinear quality mapping function. Fixing $r_l = 0.3$, we compare different r_h values, with results shown in Table 5. Nonlinear mapping significantly mitigates diversity's dominance and improves fine-tuned model performance, particularly at $r_h = 0.95$. Unlike linear methods, which improve subset quality by selecting extreme high-quality values, the nonlinear approach raises overall quality by incorporating more moderately high-quality data, aligning with its design goals.

5 Related Work

Instruction fine-tuning is widely used to refine LLMs. Early methods focused on fine-tuning with large-scale instruction datasets (Wei et al., 2022; Wang et al., 2022) manually aggregated from extensive NLP task collections (Longpre et al., 2023). With advancements in generative models, Wang et al. (2023) has led the trend of synthetic data generation (Taori et al., 2023; Ding et al., 2023; Xu et al., 2023). As Zhou et al. (2023a) found, quality and diversity are more important than quantity, driving recent efforts to cut training costs by removing low-quality and redundant data. Existing selection methods can be broadly categorized into three types (Qin et al., 2024a): quality-based, diversity-based, and model-specific importance-based selection.

Quality-based Selection Humpback (Li et al., 2023a) selects high-quality samples through an iterative self-curation process where quality predictions are produced by the fine-tuned model of each turn. Recent works typically employ a GPT-model to annotate the data quality. For example, ALPAGASUS (Chen et al., 2024) employs ChatGPT to score the accuracy of instruction data and select data according to a threshold.

Diversity-based Selection The diversity-based selection aims to deduplicate the instruction data and maximize the coverage of selected data. Recent methods typically achieve this purpose by control the nearest neighbor distance (Liu et al., 2024) or maximize the average distance between the selected data through text embedding (Wu et al., 2023). INSTAG (Lu et al., 2024) identifies semantics and intentions of instructions by tags and it assumes that a dataset is considered more diverse if it covers more individual tags.

Model-specific Importance-based Selection Importance refers to the necessity of adding one sample into training set (Qin et al., 2024a) whose indicator are typically model-specific (Xia et al., 2024; Li et al., 2024a; Hui et al., 2024; Du et al., 2023). However, this work focuses on the general data selection and emphasizes the quality and diversity of selected data.

InfoGrowth (Qin et al., 2024b) also aims to address the continuous expansion of datasets, but it primarily focuses on image data and relabeling noisy samples, making it less relevant to this paper. While InfoGrowth and DEITA consider both quality and diversity, they handle them sequentially, without combining them into a unified score. Be-

sides, previous efforts primarily aggregate all candidate data before data selection and are not experimented under the progressive instruction bank evolution task. In this paper, we propose PIBE to efficiently obtain the optimal current instruction subset with comprehensive characterization and integration of diversity and quality scores.

6 Conclusion

In this paper, we propose InsBank to address the challenge of evolving instruction subset. PIBE integrates high-quality and representative data into InsBank, striking a balance between data diversity and quality, while maintaining long-term scalability and efficiency. By leveraging a representation-based diversity score with historical information, PIBE flexibly combines diversity and quality for data selection and ranking. Experimental results show PIBE outperforms baselines, providing more optimal and adaptable instruction subsets. The orderliness of InsBank also allows users to extract tailored subsets within budget constraints, supporting cost-effective training and the ongoing refinement of LLMs. This work paves the way for more dynamic and adaptable instruction tuning strategies, enhancing both the efficiency and effectiveness of LLM development over time.

Limitations

In this work, we focus on evaluating the diversity of individual instruction data and exploring the combination of diversity and quality scores. However, achieving a more precise assessment of data quality remains a valuable direction for future research.

Ethics Statement

All of the datasets used in this study were publicly available, and no annotators were employed for our data collection. We confirm that the datasets we used did not contain any harmful content and was consistent with their intended use (research). We have cited the datasets and relevant works used in this study.

References

AI@Meta. 2024. Llama 3 model card.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard (2023-2024). https://huggingface.co/spaces/open-llm-leaderboard-old/open_llm_leaderboard.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024. Alpagasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

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.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 3029–3051. Association for Computational Linguistics.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. How abilities in large language models are affected by supervised fine-tuning data composition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 177–198. Association for Computational Linguistics.

Wei Dong, Moses Charikar, and Kai Li. 2011. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011*, pages 577–586. ACM.

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

Brendan J. Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science*, 315(5814):972–976.

Chengcheng Guo, Bo Zhao, and Yanbing Bai. 2022. Deepcore: A comprehensive library for coreset selection in deep learning. In *Database and Expert Systems Applications - 33rd International Conference, DEXA 2022, Vienna, Austria, August 22-24, 2022, Proceedings, Part I*, volume 13426 of *Lecture Notes in Computer Science*, pages 181–195. Springer.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Tingfeng Hui, Lulu Zhao, Guanting Dong, Yaqi Zhang, Hua Zhou, and Sen Su. 2024. Smaller language models are better instruction evolvers. *CoRR*, abs/2412.11231.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2024. Followbench: A multi-level fine-grained constraints following benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 4667–4688. Association for Computational Linguistics.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024a. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 7602–7635. Association for Computational Linguistics.

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

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

Yiwei Li, Jiayi Shi, Shaoxiong Feng, Peiwen Yuan, Xinglin Wang, Boyuan Pan, Heda Wang, Yao Hu, and Kan Li. 2024b. Instruction embedding: Latent representations of instructions towards task identification. *Preprint*, arXiv:2409.19680.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3214–3252. Association for Computational Linguistics.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024. What makes good data for alignment? A comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2024. #instag: Instruction tagging for analyzing supervised fine-tuning of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

OpenAI. 2022. Hello gpt-4o.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

OpenAI. 2024. Hello gpt-4o.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with GPT-4. *CoRR*, abs/2304.03277.

Yulei Qin, Yuncheng Yang, Pengcheng Guo, Gang Li, Hang Shao, Yuchen Shi, Zihan Xu, Yun Gu, Ke Li, and Xing Sun. 2024a. Unleashing the power of data

tsunami: A comprehensive survey on data assessment and selection for instruction tuning of language models. *Preprint*, arXiv:2408.02085.

Ziheng Qin, Zhaopan Xu, Yukun Zhou, Zangwei Zheng, Zebang Cheng, Hao Tang, Lei Shang, Baigui Sun, Xiaojiang Peng, Radu Timofte, Hongxun Yao, Kai Wang, and Yang You. 2024b. Dataset growth. *CoRR*, abs/2405.18347.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. 2021. Zero-offload: Democratizing billion-scale model training. *CoRR*, abs/2101.06840.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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`.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. 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), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13484–13508. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Hannaneh Hajishirzi, Noah A. Smith, and Daniel Khashabi. 2022. Benchmarking generalization via in-context instructions on 1, 600+ language tasks. *CoRR*, abs/2204.07705.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. LESS: selecting influential data for targeted instruction tuning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

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

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023a. LIMA: less is more for alignment. In *Advances in Neural*

Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023.

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

A Timeline of Instruction Datasets

Release	Dataset	Scale
2021.04	CrossFit	71M
2021.04	Natural Inst v1.0	620k
2021.09	Flan 2021	4.4M
2021.10	P3	12M
2022.04	Super-Natural Inst	5M
2022.10	FLAN 2022	15M
2022.10	MetaICL	3.5M
2022.11	xP3	81M
2022.12	Unnatural Inst	64K
2022.12	OPT-IML Bench	18M
2022.12	Self-Instruct	82K
2023.03	Alpaca	52K
2023.04	Dolly	15K
2023.04	ShareGPT	94K
2023.05	UltraChat	1.47M
2023.06	WizardLM (alpaca)	70K
2023.07	WizardLM (sharegpt)	143K
...	...	...

Figure 4: Timeline of instruction datasets (part) since 2021.04 to 2023.07.

B Momentum Responsibility Matrix

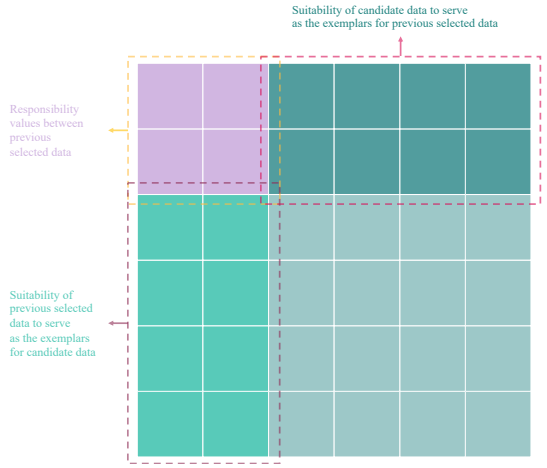

Figure 5: The structure of momentum responsibility matrix.

C Details of Implementation

Fine-grained Quality Scoring We adopt the quality annotator [3] provided by Liu et al. (2024) to score the instructions.

Representation-based Progressive Data Selection: During the PIBE data selection process, we set the momentum coefficient $\alpha = 0.3$, the momentum decaying rate $\lambda = 0.9$, the damping rate

[3] https://huggingface.co/hkust-nlp/deita-quality-scorer

$\beta = 0.5$ and the weighting coefficient $\gamma = 1$. Besides, we adopt instruction embedding (Li et al., 2024b) to encode the instructions. As for affinity propagation, we use negative euclidean distance to initialize the similarity matrix and fill the diagonal of similarity matrix with 0. Moreover, due to the high memory overhead of Affinity Propagation ($O(n^3)$), we further divided the complete set of candidates in each evolution iteration into smaller evolution batches with a batch size of 27,000 to perform PIBE. For data selection, all baselines employ the full-scale selection manner rather than the gradual selection manner to get their global optimal performance. For PIBE, we perform progressive InsBank evolution following the temporal order of dataset appearance (i.e. Self-Instruct → Alpaca → Dolly → ShareGPT → WizardLM), and take the final selected subset for model fine-tuning.

Instruction Fine-Tuning: We utilize 8 NVIDIA A100 SXM4 40GB GPUs to fine-tune LLMs. We employ LlamaFactory (Zheng et al., 2024), DeepSpeed Zero-Stage 3 (Ren et al., 2021) and fp16 precision to facilitate the training process. We adopt the Llama3-style template for Llama3-8B, Qwen-style template for Qwen2.5-7B and Mistral-style template for Mistral-7B, corresponding to "llama3" "qwen," and "mistral" template in LlamaFactory respectively. We set the effective batch size to 128 (per device train batch size=1 and gradient accumulation steps=16), training epochs to 6, learning rate to 1e-5, warmup ratio to 0.1 and maximum input length to 2048.

For trainable tokens and turns restriction, we set max tokens to 3M and max turns to 7k unless otherwise specified. For quality-controlled experiments, since all data are single-turn conversations, we set max tokens to 2M and max turns to 6k. For orderliness analysis, we set max tokens to 0.9M and max turns to 2.3k.

For AlpacaEval inference, we set temperature=0.7, top_p=0.9, top_k=40, num beams=1 and max length=512. For MT-Bench inference, we follow the default setting of FastChat[4] except for that max length is set to 512. All models adopt templates consistent with those in the training process during evaluation.

For AlpacaEval evaluation, we compare each model output with GPT-3.5 Turbo (gpt-3.5-turbo-1106) (OpenAI, 2022), because we find that when compared to text-davinci-003 (Brown et al., 2020) or GPT-4 Turbo (OpenAI, 2023), the benchmark was either too simple or too challenging, making it difficult to differentiate between models. For both AlpacaEval and MT-Bench, we employ GPT-4o (OpenAI, 2024) as annotator.

[4] https://github.com/lm-sys/FastChat/tree/main

D Correlation Analysis

We first sort the data in descending order based on the overall score and select the top 12k samples. For each sample, we assign a flag: if the sample is selected into InsBank, the flag is set to 1; otherwise, it is set to 0. We then calculate the Spearman correlation coefficients between diversity and flags, as well as between quality and flags, to investigate the contributions of diversity and quality to data selection. We restrict our analysis to the top 12k data sorted in descending order by the overall score, as we aim to focus on high-quality candidates with relatively high quality and diversity. Lower-quality candidates are excluded from the analysis since their likelihood of being selected into InsBank is inherently low.

E Quality-Controlled Subset Construction

To avoid mixing single-turn and multi-turn conversations data, as well as biases introduced by different data distributions across dataset, we sample data with quality ranging from 4.5 to 5.0 from WizardLM (alpaca), resulting in a quality-controlled subset with 19805 samples.

F Selected Data Visualization from QC-Subset

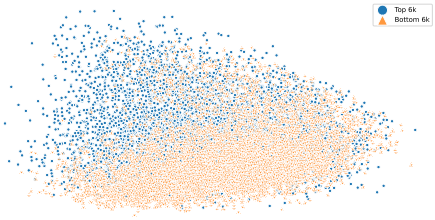

Figure 6: Selected data visualization based on quality controlled subset. The blue stars represent the most diverse data, while the orange triangles represent the least diverse data.

G Statics of Candidate Instruction Datasets

Dataset	Scale	Quality
Self-Instruct	82k	2.29
Alpaca	52k	3.59
Dolly	15k	2.76
ShareGPT (cleaned)	58k	4.03
WizardLM	70k	4.16

Table 6: Statistics of instruction datasets.

H Nonlinear Quality Mapping Function

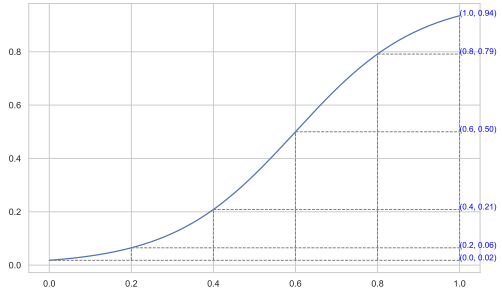

Figure 7: Visualization of nonlinear quality mapping function.

I K-Center Greedy Algorithm

Algorithm 1 K-Center Greedy

Require: data $x_i \in S$ and a budget m

Initialize $S_m = x_0$

repeat

  $u = \arg\max_{x_i \in S \setminus S_m} \min_{x_j \in S_m} d(g(x_i), g(x_j))$

  $S_m = S_m \cup \{u\}$

until $|S_m| = m$

return S_m

J Additional Analysis

J.1 Justification of Data Composition

The data composition in this work simulates the development of instruction sets. Although Self-Instruct, Alpaca, and WizardLM are related to each other, their instruction data are actually different from each other. In addition, the balance between quality and diversity during data selection is also one of the key focus of this work. By utilizing candidate instruction sets with varying quality distributions, we demonstrate that PIBE is capable of jointly considering both quality and diversity.

In this work, we focus on the efficient instruction subset evolution during the development of instruction data, thus we select Self-Instruct, Alpaca (GPT-4), Dolly, ShareGPT, and WizardLM as candidate instruction sets based on their chronological release order. These datasets collectively exhibit a trend of increasing data quality which aligns well with our scenario of data evolution.

Additionally, both quality and diversity are essential to data selection, and we have demonstrated in this paper that solely focus on one underperforms comprehensively consider both (Table 4). Therefore, high quality data alone are far from enough and including data of moderate quality to enhance data diversity is of great value. We report the quality and diversity of subsets selected by different methods in Table 2 and Table 10, showing that PIBE is able to maintain decent data quality while achieve the highest level of diversity against MoDS and DEITA. Moreover, the InsBank data distribution of each evolution step is also shown in Table 7. The final InsBank mainly consists of data from high quality datasets (ShareGPT, WizardLM), while some data from medium quality dataset (Alpaca) are also included to further enhance the diversity of InsBank. Only a limited number of samples from low-quality datasets (Self-Instruct, Dolly) are present in InsBank, showing that PIBE is able to effectively ignore low-quality samples during evolution.

Self-Instruct	Alpaca	Dolly	ShareGPT	Wizard
6000	-	-	-	-
144	5856	-	-	-
114	5695	192	-	-
9	1832	17	4142	-
3	632	17	2177	3181

Table 7: InsBank composition in different stage of InsBank evolution.

J.2 Effectiveness of Data Selection

To better demonstrate the effectiveness of data selection with high quality data, we first randomly sampled 50k data from the high quality dataset - UltraChat(Ding et al., 2023). Then, we perform DEITA and PIBE to select a 6k subset from it separately. We compare the performance of model fine-tuned with Full, DEITA and PIBE, and the results

are shown in Table 8. Both DEITA and PIBE outperform the full-data baseline, further confirming the benefits of appropriate data selection for model instruction fine-tuning. Notably, PIBE achieves the best performance, which further demonstrates its superiority.

Method	AlpacaEval	MT-Bench	IFEval
Full	20.27	5.12	32.01
DEITA	24.75	5.64	30.82
PIBE	27.86	5.73	29.26

Table 8: Data selection performance with UltraChat.

J.3 More Baselines for Comparison

In this section, we further compare PIBE with three model-specific baselines: IFD(Li et al., 2024a), IC-IFD(Hui et al., 2024) and MoDS(Du et al., 2023). As shown in Table 9 and Table 10, PIBE consistently outperforms these baselines attributing to its better balance between data quality and data diversity. For the underperformance of IFD and IC-IFD, it may due to the fact that the IFD-style metric does not guarantee the high quality of the selected data. We further check the average quality scores of subsets selected by IFD and IC-IFD, and they are significantly lower than DEITA and PIBE. For MoDS, it greatly outperforms IFD, IC-IFD due to its quality filtering strategy. However, it also handles data quality and data diversity separately, making them less balanced during the data selection process. Moreover, the augmented data selection process of MoDS is also time-consuming, making it less efficient than other data selection methods.

Method	AlpacaEval	MT-Bench	IFEval
IFD	24.50	5.01	36.57
IC-IFD	30.04	5.57	36.45
MoDS	42.83	5.83	38.01
PIBE	**44.84**	**6.23**	**40.89**

Table 9: Results of comparison between PIBE and model-specific baselines.

Metric	IFD	IC-IFD	MoDS	PIBE
Quality	3.44	3.54	5.20	5.13
Diversity	111.82	117.38	82.51	91.84

Table 10: The quality and diversity of subsets selected by different methods.

J.4 More InsBank Budgets

Method	AlpacaEval	MT-Bench	IFEval
Budget=1k			
DEITA	13.06	4.53	**37.17**
PIBE	**20.77**	**4.69**	34.05
Budget=3k			
DEITA	**43.15**	5.71	38.97
PIBE	42.79	**5.90**	**39.33**
Budget=6k			
DEITA	40.62	6.23	38.49
PIBE	**44.84**	**6.23**	**40.89**

Table 11: Results of InsBank budget of 1k and 3k.

We further conduct experiments with InsBank budget of 1k and 3k. The results in Table 11 show that increasing the data size from 1k to 3k leads to a significant improvement in model performance. However, when the data size is further increased from 3k to 6k, the performance gain becomes relatively marginal. This reflects a trend in which the model's general instruction-following ability improves rapidly with more training data but also converges quickly, which is consistent with the observations reported in (Dong et al., 2024).

J.5 More Base Models

We further conduct experiments on Llama3.2-1B, Llama3.2-3B (AI@Meta, 2024), Qwen2.5-14B (Qwen Team, 2024), the results shown in Table 12 indicate that models of sizes 1B, 3B, and 14B all greatly benefit from the instruction data and our experimental findings can further generalize to models of sizes 1B, 3B, and 14B.

J.6 Further Evaluation with More Benchmarks

We further extend the main experiments with Llama3-8B to more benchmarks (MMLU (Hendrycks et al., 2021), HellaSwag (Zellers et al., 2019), ARC (Clark et al., 2018), TruthfulQA (Lin et al., 2022), Winogrande (Sakaguchi et al., 2020) and FollowBench (Jiang et al., 2024)), and the results are shown in Table 13 and Table 14. PIBE consistently outperforms all baselines in the further evaluations, demonstrating its overall superiority.

J.7 Overlap Between Progressive Evolving and Full Data Selection

In this section, we aim to compare the overlap rates between the subsets selected by different methods

Method	AlpacaEval	MT-Bench	IFEval
Qwen2.5-14B			
base	12.19	6.89	41.01
DEITA	58.40	7.34	45.32
PIBE	**58.58**	**7.46**	**46.52**
Qwen2.5-7B			
base	14.68	6.61	40.05
DEITA	50.43	6.86	45.44
PIBE	**51.55**	**6.88**	**46.76**
Llama3-8B			
base	0.75	2.03	20.14
DEITA	43.60	6.03	38.25
PIBE	**44.84**	**6.23**	**40.89**
Llama3.2-3B			
base	0.49	1.56	17.99
DEITA	29.73	4.71	36.33
PIBE	**29.98**	**4.96**	**38.85**
Llama3.2-1B			
base	0.00	1.10	17.99
DEITA	**8.96**	3.28	31.65
PIBE	8.21	**3.39**	**31.77**

Table 12: Results of further experiment with different base models.

from the gradual manner and those from the full-scale selection manner [5].

We randomly select 40k data from the full data to obtain a subset that closely resembles the distribution of real data. We set the InsBank size here to 1k, and divided the data into four candidate subsets of 10k each to simulate the gradual manner. We compared PIBE with kNN_1 and k-Center Greedy, and perform an ablation analysis on the historical information used in PIBE. We set $\gamma = 1$, and for PIBE, we set $\alpha = 0.3$ and $\lambda = 0.9$ which aligns with the main experiment. The results are reported in Table 15. It shows that the overlap rate of PIBE exceeds that of the kNN_1 and kCenter Greedy, and the historical information also helps improve the overlap rate.

J.8 Instruction Bank Evolution

In this experiment, we investigate the performance of subsets selected by different data selection methods for model training. Following the temporal order of dataset appearance (i.e. Self-Instruct → Alpaca → Dolly → ShareGPT → WizardLM), we performed progressive InsBank evolution using PIBE and take the selected subset for model fine-tuning. The performance of the fine-tuned model across different benchmarks is shown in Figure 8.

[5]Aggregate all available candidates first and perform data selection on the full data directly.

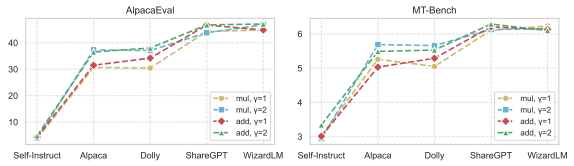

Figure 8: Model performance of different stages during InsBank evolution.

J.9 PIBE Hyper-Parameter Analysis

The damping rate β is a hyperparameter inherent to Affinity Propagation, typically set to 0.5, and we have adhered to this default setting. For the analysis of hyperparameters, we focus on examining the quality and diversity of the selected data. We compared different combinations of $\lambda = [0.9, 0.93, 0.95]$, $\alpha = [0.3, 0.5, 0.8]$, and $\gamma = [1, 2]$ in selecting InsBank. The results are shown in Figure 9. Overall, γ determines the influence of quality on data selection. As γ increases, the average quality of the selected data improves, but diversity decreases. Both λ and α determine the impact of historical information on the composition of selected data. We find that higher λ and α values generally result in lower quality but higher diversity in InsBank. This is because, according to the evolution sequence of InsBank, the quality of the data improves progressively. When the influence of historical information increases, more older data is retained in InsBank, leading to relatively lower quality and higher diversity.

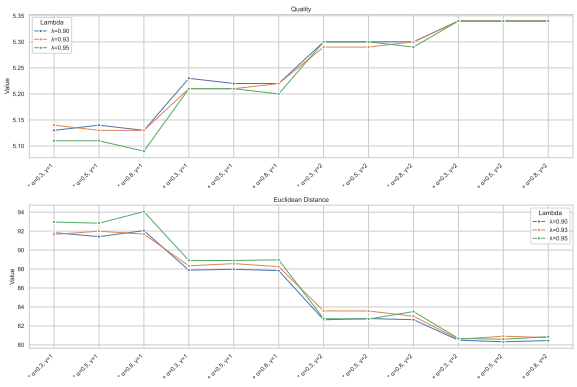

Figure 9: InsBank statistics of different hyper-parameters.

We further compare the overlap between the final InsBanks obtained with different hyperparameter. From 0 to 17, the corresponding $[\alpha, \lambda, \gamma]$ combi-

Method	MMLU	HellaSwag	ARC	TruthfulQA	Winogrande	Avg
Full	58.47	79.20	55.03	50.06	73.32	63.21
Random	60.34	**83.39**	57.88	44.69	71.88	63.63
kCenter	62.00	80.97	58.79	44.97	72.77	63.89
kNN	**64.29**	82.41	59.04	52.74	74.03	66.50
DEITA	64.15	82.95	59.90	51.81	74.43	66.64
PIBE	63.76	82.38	**61.18**	**53.55**	**75.37**	**67.24**

Table 13: Results of OpenLLM evaluation

Method	HSR					SSR					CSL
	L1	L2	L3	L4	L5	L1	L2	L3	L4	L5	CSL
DEITA	43.26	47.18	36.97	22.41	26.46	43.26	59.96	51.37	46.03	48.99	1.12
PIBE	**59.00**	**53.86**	**42.79**	**30.36**	**31.93**	**59.00**	**63.12**	**57.26**	**48.98**	**56.73**	**1.62**

Table 14: Results of FollowBench evaluation

Method	k-NN	kCenter	PIBE w/o hst	PIBE
Num	131	747	390	864

Table 15: The overlap sample number between subset selected in full-scale manner and in gradual manner. Here, PIBE w/o hst is the ablation on history information of PIBE.

nations are as follows: [0.3, 0.90, 1], [0.3, 0.93, 1], [0.3, 0.95, 1], [0.5, 0.90, 1], [0.5, 0.93, 1], [0.5, 0.95, 1], [0.8, 0.90, 1], [0.8, 0.93, 1], [0.8, 0.95, 1], [0.3, 0.90, 2], [0.3, 0.93, 2], [0.3, 0.95, 2], [0.5, 0.90, 2], [0.5, 0.93, 2], [0.5, 0.95, 2], [0.8, 0.90, 2], [0.8, 0.93, 2], [0.8, 0.95, 2]. We observe that when $\gamma = 2$, the overlap between InsBanks is generally higher compared to when $\gamma = 1$, due to the increased influence of quality. This observation is reasonable, particularly as γ continues to grows, the results increasingly resemble those of a quality-greedy data selection strategy, where the selection outcomes become fixed regardless of whether historical information is considered. When $\gamma = 1$, the influence of historical information is relatively more pronounced, resulting in significantly lower overlap rates between different InsBanks compared to when $\gamma = 2$. Additionally, we observed that when γ and λ are equal, the overlap rates of InsBanks obtained with different α values are significantly higher than those obtained when γ and α are equal but with different λ values. This indicates that λ has a greater impact on altering the influence of historical information.

J.10 Time Costs Analysis

We adhered to the data selection settings of the main experiment to compare the actual time costs of data selection between DEITA and PIBE. In this experiment, we ensure that both methods are tested under identical hardware environments. The results are shown in Table 16. It is worth noting that DEITA (full) refers to full-scale data selection, while DEITA (progressive) represents the progressive InsBank Evolution process. Additionally, the time spent loading data is also included in the total time consumption. PIBE achieves higher efficiency compared to DEITA because PIBE's data selection process is parallelized, whereas DEITA requires a sequential traversal of data to perform selection.

In practice, DEITA's data selection efficiency is primarily influenced by the number of evolution iterations and the size of InsBank. The selection time for DEITA (progressive) grows almost linearly with the number of iterations, while the total data volume has minimal impact. Additionally, as more data is selected into InsBank, the time required to select a new sample increases, as it becomes harder to find a candidate that meets the nearest neighbor similarity constraint. This implies that as the size of InsBank grows, DEITA's efficiency will further decline.

In contrast, PIBE's efficiency is unaffected by the size of InsBank due to its parallelized operations. Instead, the primary factor influencing PIBE's time consumption is the total data volume. An increase in the total data volume leads to a higher number of evolution batches, with each batch requiring approximately 1 minute to process. As a result, PIBE's total data selection time scales

linearly with the number of evolution batches.

Method	Time (hrs)
DEITA (full)	0.68
DEITA (progressive)	2.28
PIBE	0.21

Table 16: Time costs of DEITA and PIBE.

K Selected Data Quality Distribution

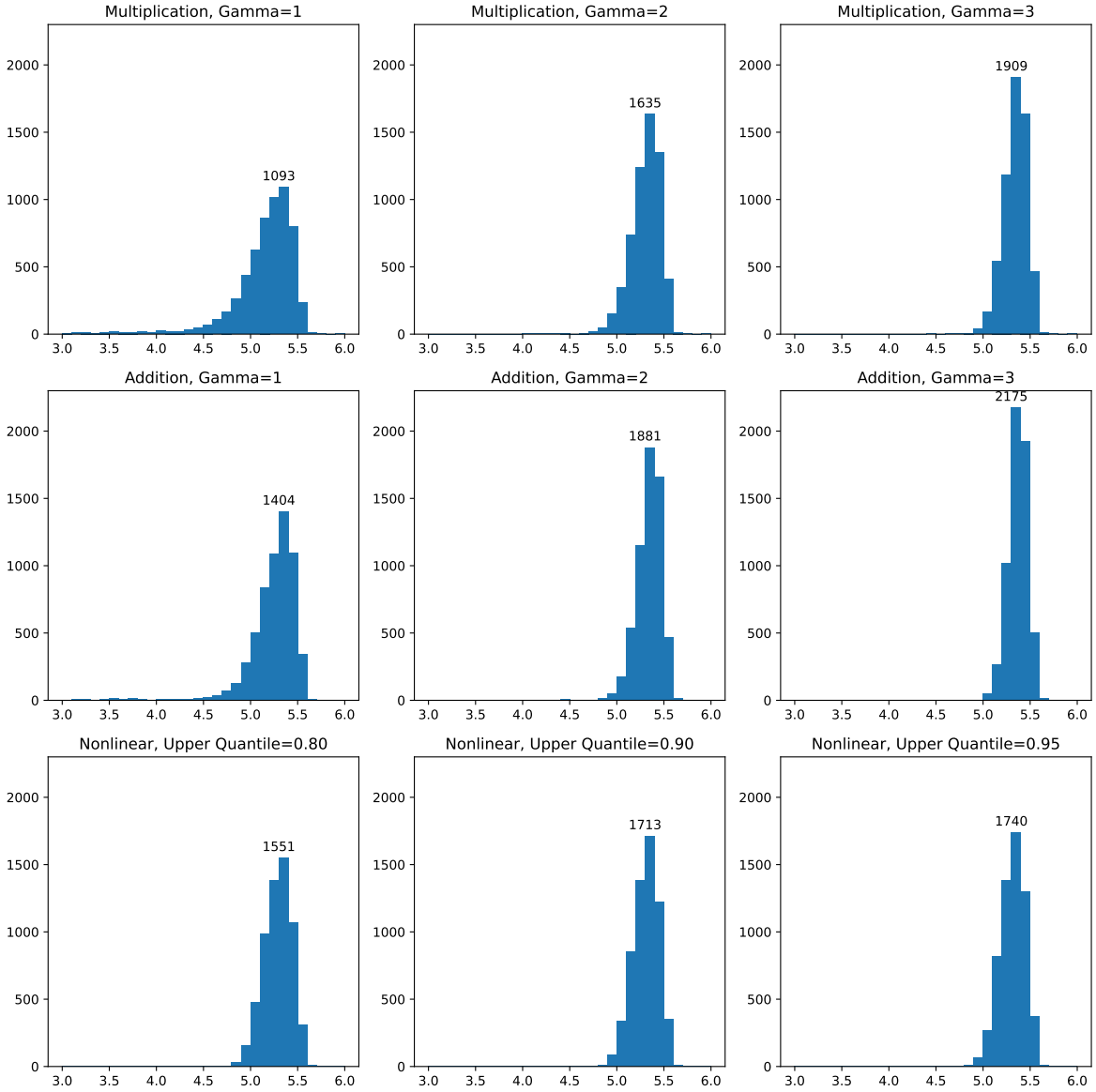

Figure 10: Selected data quality distribution of different combination approaches.