IMPROVING INFLUENCE-BASED INSTRUCTION TUNING DATA SELECTION FOR BALANCED LEARNING OF DIVERSE CAPABILITIES

Qirun Dai, Dylan Zhang, Jiaqi W. Ma, Hao Peng
University of Illinois Urbana-Champaign
{qirundai,shizhuo2,jiaqima,haopeng}@illinois.edu

ABSTRACT

Selecting appropriate training data is crucial for successful supervised instruction fine-tuning (SFT), which aims to (1) elicit strong capabilities from pretrained large language models (LLMs), and (2) achieve balanced performance across a diverse range of tasks. Algorithms based on influence estimation have shown promise in achieving (1) through estimating the contribution of each training example to model's prediction on a downstream task, but often struggle with (2). Through systematic experiments, we attribute their underperformance to an inherent bias—certain tasks intrinsically have greater influence than others. Directly comparing influence scores across different tasks would thus bias the selected data towards these tasks, hurting the LM's performance not only on other capabilities, but also, surprisingly, on the tasks for which the selected data has high influence.

We propose BIDS, a novel Data Selection algorithm that targets Influential data in a Balanced way, to address this issue. BIDS first normalizes influence scores of the training data with respect to each downstream task at an instance level. It then applies an iterative process to further balance the selection of influential training data. At each step, BIDS selects the training example that bears the highest influence on the most underrepresented capability by the currently selected data. Experimental results demonstrate that BIDS consistently outperforms state-of-the-art influence-based data selection algorithms under various budgets. Surpringly, training on a 15% subset selected by BIDS can even outperform full-dataset training with a much more balanced performance across different tasks. Our analysis further highlights the importance of both instance-level normalization and iterative optimization of selected data for balanced learning of diverse capabilities.

1 INTRODUCTION

Supervised instruction finetuing (SFT) plays a crucial role in eliciting strong capabilities from large language models (LLMs) and adapting them for various downstream tasks. Typically, a pretrained LLM is finetuned on a mixture of different datasets to achieve strong and balanced performance across a vareity of tasks (Touvron et al., 2023; Dubey et al., 2024; Jiang et al., 2023). Recent efforts have identified the importance of instruction data quality for SFT (Zhou et al., 2024), spawning many works on instruction tuning data selection (Cao et al., 2023; Chen et al., 2023; Liu et al., 2023). Among them, influence-based algorithms have shown promise. They estimate each individual training example's influence on model's prediction on a downstream task (Koh & Liang, 2017; Pruthi et al., 2020). Recent advances in influence estimation methods have enabled their scaling to LLM scales, where they have demonstrated success. These works generally select high-quality data from a large and diverse collection of instruction datasets in order to boost a specific capability of the model (Xia et al., 2024; Yang et al., 2024).

Most real-world applications call for general-purpose LLMs that have both strong and balanced capabilities on a wide range of tasks (e.g., a math tutor that excels in both math problem solving and user instruction following). However, existing influence-based data selection algorithms actually fall short in this aspect (Section 3) (Xia et al., 2024). As our analysis reveals, the core of this issue lies in the fact that when influence-based methods are applied across various tasks, the influence on some of them is inherently higher than others. As a result, directly comparing influence across tasks and selecting examples with highest influence values often result in a selected dataset biased towards the tasks with inherently higher influence. This leads to a couple of pitfalls. First, biasing the selection towards some tasks inevitably comes at the cost of the model's performance on others, making it more challenging for the LLM to achieve balanced capabilities. Second, perhaps unexpectedly, it may even hurt the model's performance on the very task that the data selection is biased towards. These issues call for an influence-based data selection algorithm designed for helping LLMs achieve balanced capabilities across a wide range of diverse tasks through instruction finetuning.

BIDS, our proposed method, addresses these challenges with two key designs. It first normalizes influence values with respect to each validation instance, enabling influence for different instances to be distributed on the same scale. Then BIDS applies an iterative selection algorithm which, at each iteration, selects the training example that provides largest influence improvement for the current selected data. This ensures that each selected example contributes most to the underrepresented tasks in the selected subset.

Our experimental results show the exceptional effectiveness of BIDS. BIDS consistently outperforms other influence-based selection algorithms, not only in terms of macro-average performance across diverse tasks, but also in most of the task-specific performance, demonstrating its effectiveness in achieving both balanced and influential data selection. Remarkably, a 15% subset by BIDS even outperforms full-dataset training, emphasizing the huge potential of selective training in fostering multi-capability learning of LLMs. Further analysis reveals the positive contributions from both the instance-level normalization and iterative selection. Investigation of the influence distribution of BIDS-selected data also provides valuable insight on how BIDS reduces the influence disparity across tasks and what might be the good properties of a balanced set of influential data.

We summarize the contribution of this work in the following three aspects:

1. We identify the problem of influence-based data selection methods in instruction tuning LLMs for learning diverse tasks, and attribute this problem to an inherent bias in cross-task influence through systematic analysis.
2. We propose BIDS, a simple and effective influence-based data selection algorithm that selects influential data for balanced capability learning.
3. Through extensive experiments, we confirm the consistent and significant improvement by BIDS, and provide valuable insights on what makes a balanced set of influential data.

2 Background and Preliminaries

Influence-based instruction tuning data selection. Estimating the influence of individual training examples on model predictions is critical for understanding model behaviors and selecting influential training data to improve model predictions. Traditional methods, including retraining-based and gradient-based approaches (Ilyas et al., 2022; Koh & Liang, 2017), have proven effective but are computationally prohibitive when scaling to LLMs. Several recent advances have sought to address these challenges by extending gradient-based approaches to scale more effectively. Given a large training dataset to select from and a validation set representing some targeted capabilities, LESS (Xia et al., 2024) models the influence between each pair of training and validation example through LoRA-based low-dimensional gradient similarity, and then selects training points with highest influence on the validation set. LOGRA (Choe et al., 2024) leverages a low-rank gradient projection algorithm to further improve the efficiency. MATES (Yu et al., 2024) formulates the pointwise data influence between each training point and the whole validation set, and uses a small data influence model to learn this pointwise influence.

Upon closer investigation, these three LLM-scale influence-based data selection methods all display similar problem formulations. They all need a validation set to represent a targeted data distribution that selected data are optimized for, and require the computation of pointwise data influence between each training instance and the validation data. In this work, we aim to extend these influence-based methods to a multi-capability instruction tuning setup. Concretely, since only LESS directly targets instruction tuning among the three LLM-scale approaches, we ground our study on the specific formulation of LESS. But we emphasize that due to the highly similar influence modeling patterns shared among these methods, the results of our work should also provide useful insight for other influence-based selection methods.

Setup. Below we introduce the problem setup and lay the groundwork for further discussion.

Definition 1 (Attribution Matrix (AM)). Assume an instruction tuning dataset $\mathcal{D}$, a validation dataset $\mathcal{V}$, which spans m diverse tasks that we want to optimize the LLM's performance for: $\mathcal{V} = \mathcal{V}_1 \cup \cdots \cup \mathcal{V}_m$, and an **influence estimation method** that can compute the influence of each training example on each validation example. We first compute all pairwise influences, yielding a $|\mathcal{D}| \times |\mathcal{V}|$ matrix $\boldsymbol{A}$. Each row of $\boldsymbol{A}$ corresponds to an individual training example, and each column a validation example. Element A_{ij} indicates the influence of the i-th example from $\mathcal{D}$ on the j-th example from $\mathcal{V}$. We dub $\boldsymbol{A}$ an **Attribution Matrix** as it reveals the overall attribution pattern from the training set to all target tasks, and each row $\boldsymbol{A}_i$ the **Influence Distribution** of the i-th training example.

Given an Attribution Matrix $\boldsymbol{A}$, our goal is to design a **data selection algorithm** that can most effectively exploit the influence information in $\boldsymbol{A}$, and select a subset $\mathcal{T}$ from $\mathcal{D}$ with size under a pre-defined budget. Finetuning the LLM on $\mathcal{T}$ is supposed to help the model achieve strong and balanced performance on all targeted tasks. Specifically, the tasks are chosen to ensure minimal overlap in terms of the capabilities they evaluate, so that the assessment range of an LLM's holistic abilities is maximized. The validation set size for each task is also kept equal to avoid bias in the selection process.

3 Exploratory Analysis

Models, datasets, and tasks. Throughout this work we use a fixed training and evaluation setting for all experiments. We consistently use the same Llama-3-8B model for both influence estimation and evaluation of selected datasets. The instruction-tuning dataset used for selection is UltraInteract (Yuan et al., 2024), a state-of-the-art, large-scale, high-quality dataset designed to enhance diverse reasoning capabilities, including mathematical reasoning, coding, and general logical inference. For the evaluation suite, we follow the original setup of UltraInteract, with seven datasets spanning five diverse capabilities. We employ HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021) for coding, GSM-Plus (Li et al., 2024) and MATH (Hendrycks et al., 2021) for math, and BigBench-Hard (BBH) (Suzgun et al., 2022) for general logical inference. We also use MMLU (Hendrycks et al., 2020) to assess the model's ability to understand and reason over world knowledge, and IFEval (Zhou et al., 2023) for the fine-grained instruction following ability. For more details about the training and evaluation setups, please refer to Appendix A.2.

For the influence estimation method, We follow the original pipeline introduced by LESS, with some case-specific modifications for the multi-task training and evaluation setups above. We also adopt the **task-wise max** selection algorithm (Appendix A.3) utilized by LESS, which, for each training example, first computes its mean influence over validation examples within the same task, followed by selecting training examples with the highest maximum influence across different tasks. We compare this algorithm against a random selection baseline, which represents the average performance of models trained on two sets of randomly selected data.

LESS fails to balance among multiple capabilities. Table 1 compares LESS against a random selection baseline. Although LESS achieves better performance in some cases, its performance is less balanced across different tasks, consistently underperforming the random baseline under both the 10% and 15% budgets in terms of macro-average score. LESS also demonstrates significant variability in performance across different tasks. For all budget levels, it significantly lags behind in BBH, while consistently outperforms the random baseline in IFEval.

The unexpectedly low and unbalanced performance of LESS may stem from the fact that it is not designed for learning multiple diverse capabilities, thus less suitable for general purpose instruction tuning. But the observations above still raise critical questions, especially given that an equal number of validation examples were used for each task during selection. This suggests a potential inherent bias in the influence values across different tasks, which could skew the selection algorithm towards certain capabilities. If the overall influence on certain task is inherently higher, then the naive task-wise max selection algorithm will naturally prioritize training examples that have high influence on these tasks, possibly at the expense of others. Further, if the oversampling of such training data

Table 1: Comparison between LESS and the random baseline. The highest performance for each task and macro-average is **bolded**. LESS only outperforms the random baseline in macro-average under the 5% budget, while lags behind under both two other budgets with imbalanced performance distribution.

Budget	Method	Coding		Logic	STEM	Math		Ins-Following	Macro Avg
		HumanEval	MBPP	BBH	MMLU	GSM-Plus	MATH	IFEval	
5%	Random	43.5	48.9	**64.8**	64.9	41.5	22.5	18.1	43.4
	LESS	**43.9**	**50.7**	62.7	**65.1**	**42.5**	**22.6**	**19.7**	**43.9**
10%	Random	**47.8**	50.6	**65.0**	**64.9**	43.9	24.0	17.8	**44.9**
	LESS	44.7	**51.3**	62.0	64.7	**44.6**	**24.3**	**19.3**	44.4
15%	Random	**48.7**	**51.9**	**65.2**	**65.1**	**45.6**	**25.0**	18.8	**45.7**
	LESS	46.5	51.0	63.2	64.6	44.9	24.9	**21.2**	45.2

doesn't bring correspondingly high improvement on this specific task, then the model would not only suffer from an imbalanced learning of the required capabilities, but also a poor overall performance.

To explore this, we actually need to examine the following two questions: 1) whether influence values differ across tasks and to what extent, and 2) whether higher influence values correlate with greater performance improvements.

Key takeaways: What causes the imbalanced failure of LESS?

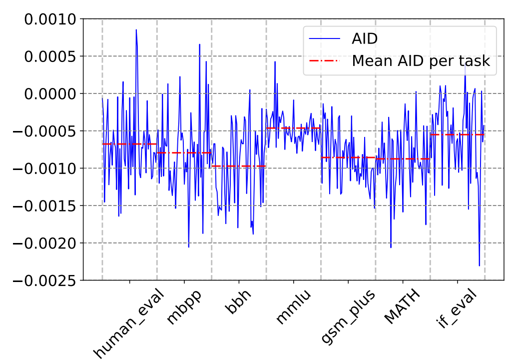


Figure 1: Unnormalized Average Influence Distribution (AID) for all seven tasks under the 10% budget, showing great inter-task and intra-task influence scale differences.

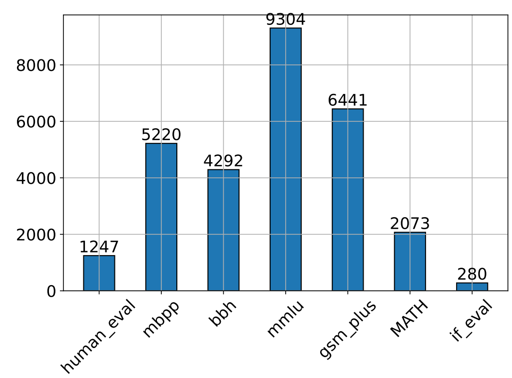


Figure 2: Task frequency with Highest Influence (THI) under the 10% budget. MMLU is obviously oversampled for in LESS-selected data.

In order to answer the questions raised in the above section, we first define the following two data analysis techniques to help us examine the influence distribution of LESS-selected data. Specifically, for any training subset $\mathcal{T} = \{\boldsymbol{t}_i\}_{i=1}^N$, we define the following two analysis objects:

- **Average Influence Distribution (AID)**. The AID of $\mathcal{T}$ is defined as the average of influence distributions of all the training points inside, i.e., AID $\triangleq \frac{1}{N}\sum_{i=1}^N \boldsymbol{A}_i$
- **Task frequency with Highest Influence (THI)**. For each training point $\boldsymbol{t}_i \in \mathcal{T}$, if its average influence on task j is the highest across all tasks, i.e., $j = \arg\max_{k=1,\ldots,m}\{\sum_{\boldsymbol{v}_j \in \mathcal{V}_k} \boldsymbol{A}_{ij}\}$, then the THI frequency for task j increases by one.

Our AID analysis of the whole UltraInteract dataset (Figures 1) reveals both task- and instance-level discrepancies. Notably, MMLU exhibits the highest average influence, nearly 50% less negative than BBH. Moreover, discrepancies of average influence inside the same task can exceed the largest instance-wise average influence by 2.5 times. These results answer our question 1) by confirming that the scales of influence values indeed differ significantly across various tasks.

Algorithm 1 BIDS: Iterative Selection Favoring Underrepresented Tasks

Input: $\mathcal{D}$: a set of training examples, each of which is represented by its normalized influence distribution; B: the number of examples to be selected; $\mathcal{V}$: the set of validation examples.
Initialization: $\mathcal{T} = \varnothing, \mathcal{D} = \{\boldsymbol{A}_i\}_{i=1}^{N}$
while $|\mathcal{T}| < B$ **do**
  $\boldsymbol{A}^* = \arg\max_{\boldsymbol{A}_i \in \mathcal{D}} \max_{j=1}^{|\mathcal{V}|} \left\{ \boldsymbol{A}_{ij} - \frac{1}{|\mathcal{T}|} \sum_{\boldsymbol{A}_k \in \mathcal{T}} \boldsymbol{A}_{kj} \right\}$
  $\mathcal{T} = \mathcal{T} \cup \{\boldsymbol{A}^*\}$
  $\mathcal{D} = \mathcal{D} \setminus \{\boldsymbol{A}^*\}$
end while
Return: $\mathcal{T}$: selected training examples.

Further, the THI analysis of LESS-selected data (Figure 2) validates that the scale differences indeed make the selection algorithm of LESS disproportionately prioritize certain tasks over others. Specifically, MMLU has the highest frequency of being the most influential task, which is consistent with the observations in Figure 1 that MMLU has the highest task-level average influence. However, the oversampling of MMLU training data does not translate into proportionally better performance—LESS often underperforms the random baseline on MMLU, despite targeting this task with its selected data. This crucial observation underscores that a higher influence score does not necessarily imply a larger performance improvement, and, paradoxically, it may hinder the learning of other necessary capabilities. Thus, we answer the question 2) by concluding that the inherent difference in the influence value scales across tasks is indeed a harmful bias that can severely undermine the performance of the data selection algorithm employed by LESS.

4 BIDS: Selecting Influential Data for Balanced Capability Learning

To address the imbalanced learning issue caused by the inherent bias in multi-task influence, we propose BIDS, a simple and effective Data Selection algorithm aiming to select Influential training data in a Balanced way. Given the Attribution Matrix (AM) $\boldsymbol{A}$, BIDS applies two crucial operations to $\boldsymbol{A}$ sequentially: **instance-level normalization**, and **iterative selection favoring underrepresented tasks**.

Instance-level normalization. The analyses presented in the above section reveal that validation instances exhibit substantial variability in the scales of their influence values. If left unaddressed, such variability introduces bias into influence-based data selection processes, thus hindering the balanced optimization of model performance across tasks. To mitigate this, we implement an instance-level normalization technique. From the perspective of Attribution Matrix, this technique applies a column-wise normalization in order to align the influence scores on a consistent scale. Specifically, for each validation instance $\boldsymbol{v}_j$, the influence of each training example $\boldsymbol{t}_i$ is normalized as below:

$$\boldsymbol{A}_{ij}^{\text{norm}} = \frac{\boldsymbol{A}_{ij} - \mu_j}{\sigma_j}$$

where μ_j and σ_j are the sample mean and standard deviation of all values in column j of the AM. After such normalization, the influence scores of different columns should be distributed on the same scale, with scores of similar rankings in different columns having similar values.

One potential issue with such a normal standardization technique is that it may not work sufficiently well when the distribution of unnormalized data differs much from an approximate normal distribution. But we show that after normalization, most of the columns in the AM indeed approximate a standard normal distribution, justifying the use of this technique for instance-level normalization of influence scores. Please refer to Appendix A.4 for more details.

Iterative selection favoring underrepresented tasks. To further ensure a balanced distribution of influence across the selected data, we propose an iterative greedy selection algorithm (detailed in Algorithm 1). The algorithm begins with an empty set of selected instances and, at each iteration, selects the training instance that provides the largest improvement in influence on any validation instance, ensuring that each newly added instance contributes maximally to underrepresented tasks or capabilities. The iterative process continues until the selection budget is fully utilized. This approach essentially differs from the naive selection algorithm used in LESS, which only scores each training instance independently and then selects the top-ranked instances, by considering the interactions of influence distributions among different selected instances and promoting the balance of overall influence distribution of the selected dataset.

5 Experiments

5.1 Experimental Setups

Basic setups. We follow the experimental setup outlined in Section 3, including the same set of LLMs, datasets, tasks, and influence estimation implementations. To extend the scope of the study, in this section we mainly introduce additional baselines and selection algorithms that are not covered in the previous exploratory analysis.

Baselines. For a more thorough comparison, we evaluate BIDS against several other algorithms tailored for influence-based selection, extending beyond the original task-wise max algorithm in the exploratory analysis. Please refer to Appendix A.3 for the mathematical definition of all the baseline selection algorithms in this work. In this section, we additionally include:

- **Instance-wise max.** For each training example, this algorithm first computes the maximum of its influence values over all the validation instances and assigns it as the overall score. Then it selects training examples with highest overall scores.
- **Sum.** Similar to the instance-wise max algorithm, this algorithm also assigns an overall score to each training example and then selects the highest, but by computing the summation of its influence instead of maximum.

5.2 Main Results

Table 2: Comparison between BIDS and other selection algorithms. The model is trained on the selected data for two epochs unless otherwise specified. The task-specific and macro-average performance of BIDS is **bolded** if it outperforms at least two other baselines, and further **underlined** if it outperforms all other four baselines. When scaling the training of BIDS to four epochs, it outperforms full-dataset training with both one and four epochs, showing its consistently strong and balanced performance.

Budget	Method	Coding		Logic	STEM	Math		Ins-Following	Macro Avg
		HumanEval	MBPP	BBH	MMLU	GSM-Plus	MATH	IFEval	
5%	Random	43.5	48.9	64.8	64.9	41.5	22.5	18.1	43.4
	Task-max	43.9	50.7	62.7	65.1	42.5	22.6	19.7	43.9
	Sum	45.6	51.9	63.6	64.8	42.4	21.3	20.1	44.2
	Instance-max	43.9	52.1	63.2	65.0	42.6	22.3	20.6	44.2
	BIDS	**45.6**	**51.0**	**64.3**	**64.9**	42.1	**22.9**	**21.4**	**44.6**
10%	Random	47.8	50.6	65.0	64.9	43.9	24.0	17.8	44.9
	Task-max	44.7	51.3	62.0	64.7	44.6	24.3	19.3	44.4
	Sum	45.6	51.6	61.6	64.6	43.8	23.7	21.0	44.6
	Instance-max	46.5	47.3	64.6	65.0	44.1	24.7	22.8	45.0
	BIDS	**48.2**	50.4	**65.1**	**64.9**	**45.1**	**25.1**	**23.4**	**46.0**
15%	Random	48.7	51.9	65.2	65.1	45.6	25.0	18.8	45.7
	Task-max	46.5	51.0	63.2	64.6	44.9	24.9	21.2	45.2
	Sum	48.2	51.0	62.6	64.6	44.8	24.0	19.3	44.9
	Instance-max	47.4	48.1	63.2	65.0	45.8	25.1	20.3	45.0
	BIDS	**49.1**	50.7	**63.7**	**64.6**	**45.8**	**26.2**	**22.6**	**46.1**
	BIDS (epochs=4)	**50.0**	53.0	**64.4**	**64.7**	47.0	26.9	**23.4**	**47.1**
100%	Full (epochs=1)	52.6	53.6	65.5	64.1	47.2	27.9	17.5	46.9
	Full (epochs=4)	48.2	54.4	59.2	63.1	51.5	32.3	17.9	46.7

Comparison under the same budget. We first compare BIDS with various other influence-based selection algorithms under the same selection budget, as is shown in Table 2. Across various budgets spanning 5%, 10% and 15%, we find BIDS consistently outperforms other algorithms in terms of the macro-average score across all seven benchmarks. Moreover, when comparing the specific performance on each benchmark separately, we find BIDS outperforms at least two baselines on 6/7 benchmarks, and outperforms all the four baselines on more than half of these benchmarks under most of the selection budgets. These two findings together show that BIDS indeed helps improve the LLM's performance on multiple tasks in a significant but also balanced way.

Comparison with full-dataset training. As shown in the last three lines in Table 2, training on a 15% subset selected by BIDS over four epochs consistently outperforms full-dataset training. Further analysis on benchmark-specific performance reveals that BIDS achieves superior performance by maintaining balanced proficiency across six reasoning-related tasks while significantly improving instruction-following capabilities. These results demonstrate that BIDS not only excels in selecting influential and balanced data, but also that full-dataset training may not always be optimal for fostering robust, multi-capability learning in LLMs. This finding highlights the potential for training on selective subsets to offer more efficient and effective learning effects for LLMs.

6 Additional Analysis

In this section we mainly discuss the respective contribution of the two additional techniques introduced in BIDS: instance-level normalization, and iterative selection favoring underrepresented tasks. Through extensive ablation experiments and data analysis, we not only validate the respective positive contribution of these two techniques, but also explore how they affect the distribution of selected data specifically, providing valuable insight on what might be the good properties of a balanced set of influential data.

6.1 Both Normalization and Iterative Selection contribute positively

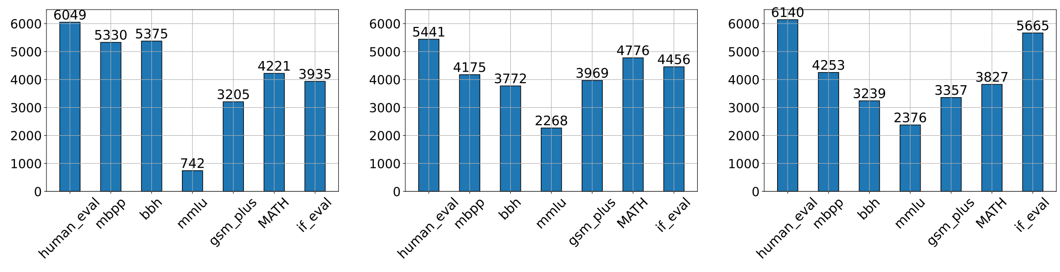
Table 3: Respective contribution of normalization and iterative Selection. The highest performance for each task and macro-average is **bolded**. "Unnormalized" refers to algorithm (1), and "Normalized" and BIDS refer to (2) and (3) respectively. It shows both normalization and iterative selection make positive contribution to model performance.

Budget	Method	Coding		Logic	STEM	Math		Ins-Following	Macro Avg
		HumanEval	MBPP	BBH	MMLU	GSM-Plus	MATH	IFEval	
5%	Unnormalized	43.9	**52.1**	63.2	**65.0**	**42.6**	22.3	20.6	44.2
	Normalized	45.6	**52.1**	62.5	64.8	42.5	22.5	20.1	44.3
	BIDS	**45.6**	51.0	**64.3**	64.9	42.1	**22.9**	**21.4**	**44.6**
10%	Unnormalized	46.5	47.3	64.6	65.0	44.1	24.7	22.8	45.0
	Normalized	47.4	48.4	64.6	**65.1**	**45.4**	**25.2**	23.0	45.6
	BIDS	**48.2**	**50.4**	**65.1**	64.9	45.1	25.1	**23.4**	**46.0**
15%	Unnormalized	47.4	48.1	63.2	**65.0**	**45.8**	25.1	20.3	45.0
	Normalized	47.4	50.1	**64.9**	**65.0**	45.6	26.0	20.8	45.7
	BIDS	**49.1**	**50.7**	63.7	64.6	**45.8**	**26.2**	**22.6**	**46.1**

Algorithms to compare. In the ablation experiments, we compare the following three selection approaches: (1) Instance-wise max algorithm applied to the original, unnormalized Attribution Matrix computed by LESS; (2) Instance-wise max algorithm applied to the Attribution Matrix after instance-level normalization; (3) BIDS. These three algorithms all originate from the naive instance-wise max approach, but from (1) to (2) the technique of normalization is applied, and from (2) to (3) the iterative selection algorithm is further employed. Therefore, the comparison of these three approaches enables us to clearly see the respective contribution of the two techniques of interest.

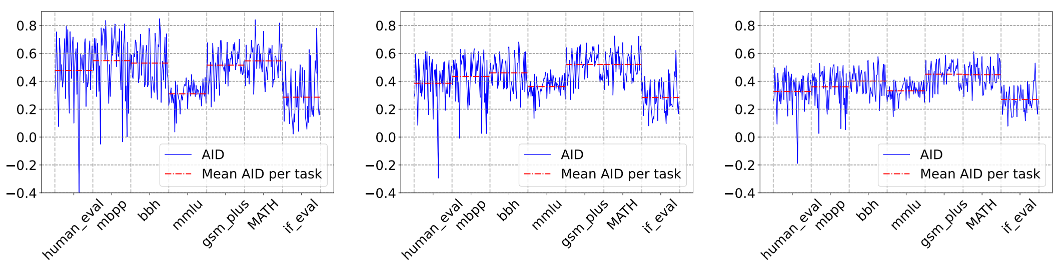
Conclusions. From Table 3, we observe that instance-level normalization alone can already consistently boost the overall performance of selected data under various selection budgets. And applying the iterative selection not only further elevates the macro-average score, but also improves the balance of model capability across diverse tasks. These two observations confirm that the remarkable performance of BIDS comes from the compound positive effects from both normalization and iterative greedy selection.

6.2 Effects of BIDS on influence distribution of selected data



(a) Unnormalized (b) Normalized (c) BIDS

Figure 3: Comparative analysis of THI under the 10% budget. Both Normalized and BIDS have more balanced task frequencies compared with Unnormalized.



(a) Unnormalized (b) Normalized (c) BIDS

Figure 4: Comparative analysis of normalized AID under the 10% budget. From Unnormalized to Normalized to BIDS, the disparity among different tasks and instances in AID gradually diminishes, with both decreasing upper bounds and increasing lower bounds.

After confirming the positive contribution from both of the two components of BIDS, we then proceed to explore how they respectively affect the influence distribution of selected data, and whether such effects can provide some insight on why BIDS advances balanced learning of diverse capabilities.

In this section we still compare among the data selected by the three algorithms mentioned in above section: (1) Unnormalized; (2) Normalized; (3) BIDS, based on the two types of data analysis techniques defined in Section 3:

1. **Average Influence Distribution (AID)** across all the 350 validation examples. For better comparison among different algorithms, here we unify the AID analysis with influence values after instance-level normalization.
2. **Task frequency with Highest Influence (THI)** across all seven tasks. Here we use a slightly different definition of task frequency than the one defined in Section 3, by replacing task-wise average influence with instance-wise influence, since the three algorithms we are comparing now are all built upon the instance-wise max approach. Concretely, for each training point $\boldsymbol{t}_i$, if its influence on validation point $\boldsymbol{v}_k$ is the highest across all $|\mathcal{V}|$ validation instances and $\boldsymbol{v}_k \in \mathcal{V}_j$, then the THI frequency for task j increases by one.

Observing the THI analysis results (Figure 3) of these three sets of selected data, we notice that after normalization, the task frequency distribution becomes much more balanced. The frequencies for tasks such as MMLU, GSM-Plus, MATH and IFEval all increase by a great extent, while those for BBH and the two coding tasks decrease. This is fairly surprising when compared with the experimental results in Table 3, where algorithms (2) and (3) actually show improvements both in tasks with decreased and increased THI frequencies compared with (1). This observation suggests that a balanced selection of influential data may improve data efficiency not only by allocating more budget to capabilities that is underrepresented, but also reducing the redundancy in over-represented capabilities.

Additionally, the AID results (Figure 4) provide further insights from a different perspective. From algorithm (1) to (2) to (3), we observe a gradual reduction in the disparity of average influence across tasks. This change unfolds in two interesting ways:

1. The upper bounds of average influence diminish almost for each task. Despite generally lower influence scores across these evaluation tasks, the performance of BIDS improves consistently compared to both the normalized and unnormalized instance-wise max selection algorithms. We remark that this observation actually reveals a limitation of the first-order linearity assumption by the influence estimation method of LESS: simply selecting high-influence points using a top-K algorithm increases the average influence distribution on almost all tasks, but their effectiveness doesn't linearly add up, thus not necessarily improving task-level or overall performance.
2. The lower bounds of average influence increase, especially for tasks with validation instances that have exceptionally low influence values, such as HumanEval and MBPP. This observation again suggests the effectiveness of one of BIDS's key motivations: improving the model's overall performance by enhancing the capabilities that are most underrepresented in the current selected data.

7 RELATED WORK

Data Selection for LLM Instruction Tuning. Since the pioneering work LIMA (Zhou et al., 2024) showed that a mere 1000 carefully curated high-quality instruction data can already lead to significant performance improvement, many works have been exploring automatic data selection pipelines guided by different metrics. Quality-guided selection mostly defines the quality for each data point based on natural language indicators (Cao et al., 2023), quality scores from strong evaluators such as GPT-4 (Chen et al., 2023; Parkar et al., 2024), or principled metrics derived from various learning dynamics (Kang et al., 2024; Mekala et al., 2024; Xia et al., 2024; Choe et al., 2024). Diversity-guided methods usually perform clustering over certain informative representation of each data point (Yang et al., 2024), and also take inspiration from traditional core-set selection approaches (Das & Khetan, 2023). Both of these dimensions have been proved effective for instruction tuning LLMs (Bukharin & Zhao, 2023; Liu et al., 2023), and we remark that our method BIDS considers both quality and diversity metrics through its iterative selection algorithm based on influence distributions.

Influence Estimation. Influence estimation has long been an important type of data attribution method, which can be classified into gradient-based and retraining-based approaches (Hammoudeh & Lowd, 2024; Ko et al., 2024). Gradient-based influence estimation focuses on the gradient trace of each training point, and assesses the gradient alignment between training and validation examples (Koh & Liang, 2017; Pruthi et al., 2020). Retraining-based estimation usually starts by training models on various subsets, and then inspects how the performance of these models changes when a training example is added to these subsets (Ghorbani & Zou, 2019; Ilyas et al., 2022; Park et al., 2023). Recently both lines of influence estimation works have been extended to LLM-scale applications, covering various aspects including pretraining (Engstrom et al., 2024; Yu et al., 2024; Choe et al., 2024) and instruction tuning (Xia et al., 2024; Liu et al., 2024).

8 CONCLUSION

In this work, we introduce BIDS, an influence-based instruction tuning data selection algorithm specifically designed for balanced learning of multiple diverse capabilities. Motivated by the observation of an inherent bias in influence across various tasks, BIDS first applies instance-level normalization to a given Attribution Matrix. Together with an iterative selection algorithm favoring underrepresented tasks, BIDS consistently outperforms various selection algorithms as well as full-dataset training with much more balanced performance. Our analysis further provides insight on the good properties of an influential dataset with balanced capabilities.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

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

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

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

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Sang Keun Choe, Hwijeen Ahn, Juhan Bae, Kewen Zhao, Minsoo Kang, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, et al. What is your data worth to gpt? llm-scale data valuation with influence functions. *arXiv preprint arXiv:2405.13954*, 2024.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Logan Engstrom, Axel Feldmann, and Aleksander Madry. Dsdm: Model-aware dataset selection with datamodels. *arXiv preprint arXiv:2401.12926*, 2024.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pp. 2242–2251. PMLR, 2019.

Zayd Hammoudeh and Daniel Lowd. Training data influence analysis and estimation: A survey. *Machine Learning*, 113(5):2351–2403, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

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

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data. *arXiv preprint arXiv:2202.00622*, 2022.

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

William B. Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into hilbert space. *Contemporary mathematics*, 26:189–206, 1984. URL `https://api.semanticscholar.org/CorpusID:117819162`.

Feiyang Kang, Hoang Anh Just, Yifan Sun, Himanshu Jahagirdar, Yuanzhi Zhang, Rongxing Du, Anit Kumar Sahu, and Ruoxi Jia. Get more for less: Principled data selection for warming up fine-tuning in llms. *arXiv preprint arXiv:2405.02774*, 2024.

Myeongseob Ko, Feiyang Kang, Weiyan Shi, Ming Jin, Zhou Yu, and Ruoxi Jia. The mirrored influence hypothesis: Efficient data influence estimation by harnessing forward passes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26286–26295, 2024.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers. *arXiv preprint arXiv:2402.19255*, 2024.

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

Zikang Liu, Kun Zhou, Wayne Xin Zhao, Dawei Gao, Yaliang Li, and Ji-Rong Wen. Less is more: Data value estimation for visual instruction tuning. *arXiv preprint arXiv:2403.09559*, 2024.

Dheeraj Mekala, Alex Nguyen, and Jingbo Shang. Smaller language models are capable of selecting instruction-tuning training data for larger language models. *arXiv preprint arXiv:2402.10430*, 2024.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. *arXiv preprint arXiv:2303.14186*, 2023.

Ritik Sachin Parkar, Jaehyung Kim, Jong Inn Park, and Dongyeop Kang. Selectllm: Can llms select important instructions to annotate? *arXiv preprint arXiv:2401.16553*, 2024.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33: 19920–19930, 2020.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

Yu Yang, Siddhartha Mishra, Jeffrey N Chiang, and Baharan Mirzasoleiman. Smalltolarge (s2l): Scalable data selection for fine-tuning large language models by summarizing training trajectories of small models. *arXiv preprint arXiv:2403.07384*, 2024.

Zichun Yu, Spandan Das, and Chenyan Xiong. Mates: Model-aware data selection for efficient pretraining with data influence models. *arXiv preprint arXiv:2406.06046*, 2024.

Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, et al. Advancing llm reasoning generalists with preference trees. *arXiv preprint arXiv:2404.02078*, 2024.

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

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

A Appendix

A.1 Influence Estimation Pipeline of LESS

In this section we briefly introduce the influence estimation pipeline of LESS. For more detailed motivation and step-by-step mathematical deduction, we suggest referring to (Xia et al., 2024).

Assume a model $\mathcal{M}_s$ which scores and selects data, and another model $\mathcal{M}_t$ which is trained on the selected data. For a training dataset $\mathcal{D}$ and validation dataset $\mathcal{V}$, LESS formulates the pairwise influence between each training point $\boldsymbol{t}_i \in \mathcal{D}$ and validation point $\boldsymbol{v}_j \in \mathcal{V}$ using the following two steps:

Step 1: Warmup training with LoRA. LESS first trains $\mathcal{M}_s$ on a random subset $\mathcal{D}_{\text{warmup}} \subset \mathcal{D}$ for N epochs using LoRA (Hu et al., 2021), checkpointing the model after each epoch to store LoRA parameters $\{\boldsymbol{\theta}_t\}_{t=1}^N$.

Step 2: Gradient computation and projection. For each checkpoint $\boldsymbol{\theta}_t$ of LoRA-trained $\mathcal{M}_s$, LESS computes the SGD gradient of validation point $\boldsymbol{v}_j$, and further uses random projection (Johnson & Lindenstrauss, 1984; Park et al., 2023) to project the gradient to a tractable lower dimension. The resulting projected gradient is denoted as $\nabla l(\boldsymbol{v}_j; \boldsymbol{\theta}_t)$. LESS also computes and projects the gradient of training point $\boldsymbol{t}_i$, but uses the Adam gradient defined as follows:

$$\Gamma(\boldsymbol{t}_i, \boldsymbol{\theta}_t) \triangleq \frac{\boldsymbol{m}^{t+1}}{\sqrt{\boldsymbol{v}^{t+1} + \epsilon}}$$

where $\boldsymbol{m}^{t+1}$ and $\boldsymbol{v}^{t+1}$ are the first and second moments in the parameter update rule for Adam optimizer.

Step 3: Gradient matching and influence calculation. Finally, LESS employs the following cosine-similarity-based approach to calculate the similarity between the gradient of each training and validation example, accumulated over all the warmup training epochs:

$$\text{Inf}_{\text{Adam}}(\boldsymbol{t}_i, \boldsymbol{v}_j) \triangleq \sum_{t=1}^{N} \overline{\eta_t} \cos(\nabla l(\boldsymbol{v}_j; \boldsymbol{\theta}_t), \Gamma(\boldsymbol{t}_i, \boldsymbol{\theta}_t))$$

where $\overline{\eta_t}$ is the average learning rate in the t-th epoch.

A.2 Details of training and evaluation setups

Based on the LESS pipeline described above, we further introduce the implementation details of the training and evaluation setups in this work. All the experiments are carried out on 2 80GB H100 GPUs.

Training Details. We basically follow the same set of hyperparameters as LESS when training both M_s and M_t. Specifically, a batch size of 128 is used throughout all the training processes in this work, along with a learning rate scheduler with linear warm-up, cosine decay, and a peak learning rate of 2×10^{-5}. For the influence estimation pipeline, we consistently conduct the warmup training of M_s using four epochs and the full training set. For gradient computation and projection, we uniformly sample 50 validation examples from the test split of each of the seven evaluation tasks, leading to a total of 350 validation examples. The projection dimension is set as 8192 for all the training and validation instances. For training M_t on the selected data, we consistently train for two epochs if not otherwise specified.

Both the warmup training for influence estimation and the training on selected data are carried out with LoRA. The LoRA configurations are kept the same throughout the experiments, with a rank of 128, an α value of 512, a dropout rate of 0.1, and LoRA matrices being applied to all the attention modules.

Evaluation Details. We follow the evaluation convention of UltraInteract (Yuan et al., 2024) by using greedy decoding (i.e., temperature $= 0$) for all the evaluation tasks except for IFEval, where we use temperature $= 0.7$ and take the median result of three random seeds due to the high variability of this task.

A.3 Mathematical definition of baseline selection algorithms

In this section, we provide the mathematical definition of all the three baseline selection algorithms that are used in this work. They share the same framework of first assigning an overall score s_i to each training example t_i and then selecting examples with the highest overall scores, and only differ in the specific definition of s_i.

Task-wise Max: $s_i \triangleq \max_{k=1,\ldots,m}\{\sum_{\boldsymbol{v}_j \in \mathcal{V}_k} \boldsymbol{A}_{ij}\}$.

Instance-wise Max: $s_i \triangleq \max_{j=1,\ldots,|\mathcal{V}|}\{\boldsymbol{A}_{ij}\}$.

Sum: $s_i \triangleq \sum_{j=1}^{|\mathcal{V}|} \boldsymbol{A}_{ij}$.

A.4 Effect of Normal standardization on attribution matrix

In this section we aim to justify the application of normal standardization to Attribution Matrix (AM). Specifically, we randomly select five validation instances (i.e., five columns in AM) from each task, and compare their empirical distributions after normalization with a standard normal distribution. The results show that almost all of the columns sampled approximate a standard normal distribution after the instance-level normalization, which justifies the use of normal standardization as the normalization method in BIDS.

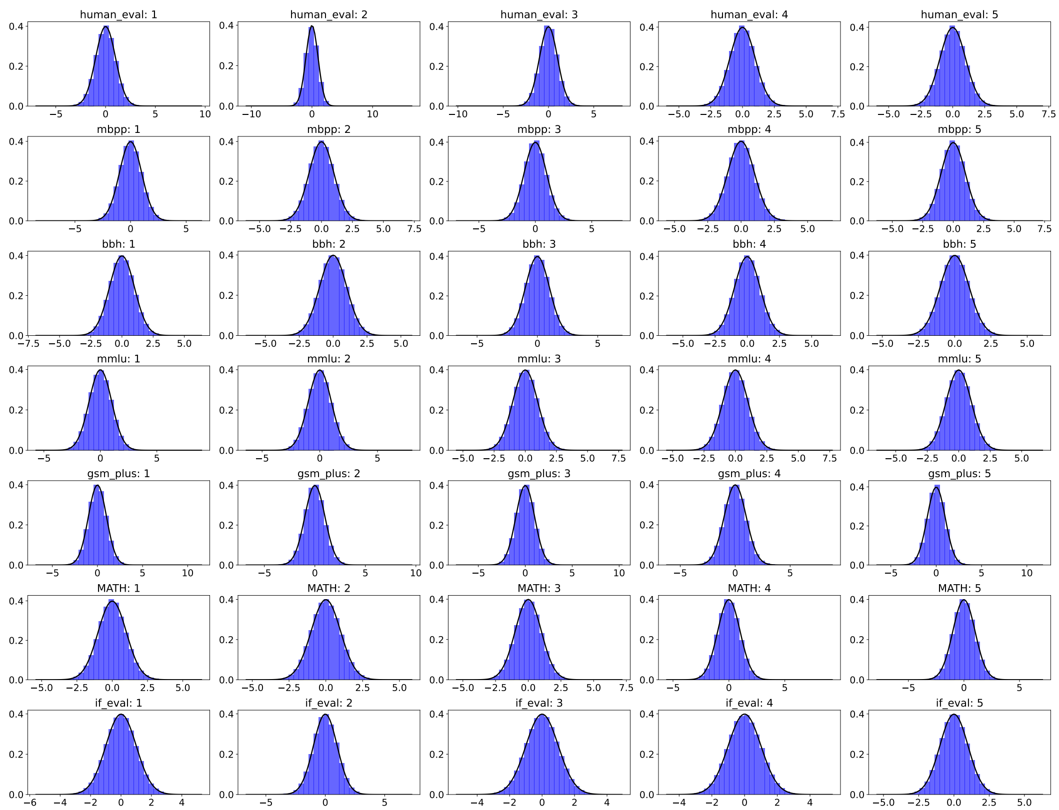


Figure 5: The effect of normal standardization. Five AM columns are sampled for each task. Most of the columns in the AM indeed approximate a standard normal distribution after normal standardization.