SHED: Shapley-Based Automated Dataset Refinement for Instruction Fine-Tuning

Yexiao He[1] **Ziyao Wang**[1] **Zheyu Shen**[1] **Guoheng Sun**[1] **Yucong Dai**[2]
Yongkai Wu[2] **Hongyi Wang**[3] **Ang Li**[1]
[1]University of Maryland [2]Clemson University [3]Rutgers University
{yexiaohe,ziyaow,zyshen,ghsun,angliece}@umd.edu
{yucongd,yongkaw}@clemson.edu
hongyi.wang.001@rutgers.edu

Abstract

The pre-trained Large Language Models (LLMs) can be adapted for many downstream tasks and tailored to align with human preferences through fine-tuning. Recent studies have discovered that LLMs can achieve desirable performance with only a small amount of high-quality data, suggesting that a large portion of the data in these extensive datasets is redundant or even harmful. Identifying high-quality data from vast datasets to curate small yet effective datasets has emerged as a critical challenge. In this paper, we introduce SHED, an automated dataset refinement framework based on Shapley value for instruction fine-tuning. SHED eliminates the need for human intervention or the use of commercial LLMs. Moreover, the datasets curated through SHED exhibit transferability, indicating they can be reused across different LLMs with consistently high performance. We conduct extensive experiments to evaluate the datasets curated by SHED. The results demonstrate SHED's superiority over state-of-the-art methods across various tasks and LLMs; notably, datasets comprising only 10% of the original data selected by SHED achieve performance comparable to or surpassing that of the full datasets.

1 Introduction

The development of LLMs marks a major leap in machine learning, transforming how we approach natural language processing (NLP) and artificial intelligence (AI) research [1, 2, 3, 4, 5]. LLMs such as GPT-3 [2], Mistral [6], and LLaMA/LLaMA2 [3, 4] highlight the benefits of pre-training on large and diverse mixtures of data corpora, empowering these LLMs with a wealth of knowledge[7, 8]. Moreover, one of the pivotal strengths of LLMs lies in their adaptability to specific tasks through fine-tuning. Fine-tuning, a process that involves adapting LLMs to one or multiple task-specific datasets, enables the pre-trained LLM to acquire task-specific information. Furthermore, it facilitates the alignment of LLMs to more accurately follow human instructions through fine-tuning on a dataset comprised of instructions paired with appropriate responses[9], which is known as instruction tuning.

However, fine-tuning LLMs also raises challenges. A primary concern is that noisy data or harmful instances in the fine-tuning dataset can significantly degrade the performance of pre-trained LLMs [10]. While many works have developed large and diverse datasets for fine-tuning purposes, recent research suggests that meticulously curated datasets of high quality, even if smaller in size, can be more effective in harnessing the full potential of LLMs [11, 12, 13]. Indiscriminately increasing the volume of data can lead to ineffective performance improvements and might even deteriorate LLM performance due to the introduction of noisy and harmful instances. Additionally, for instruction tuning, the LLM has already learned the necessary knowledge in the pre-training stage. The dataset used in the fine-tuning stage merely aims to better align the LLM to follow human instructions, indicating that this process does not necessitate extensive data [14]. Furthermore, fine-tuning LLMs on extensive datasets incurs significant computational costs. The necessity for considerable GPU

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3153

resources presents a critical challenge [15]. Only researchers and institutions equipped with sufficient computing resources can perform such tasks, limiting broader applications and progress within the LLM community. Consequently, there is a pressing need to design a novel method for curating small and high-quality datasets that enable efficient fine-tuning.

Previous efforts have employed various methods such as curation or generation through manual efforts or commercial LLMs [11, 16], identifying subsets from larger datasets via training dynamics or estimating marginal contributions [17, 18]. Most current methods for data selection neglect the potential influence that different combinations of samples can have on model performance. The Shapley value [19], introduced in cooperative game theory, provides a method for fairly evaluating the contribution of each participant by examining all possible combinations and their effects on the overall result. This principle has also been utilized in machine learning to assess the impact of individual data points within a given dataset [20]. The Shapley value can serve as a criterion to refine one or more large datasets to extract high-quality data points, enabling the curation of a smaller yet high-quality dataset. This method not only facilitates the selection of impactful data but also considers the effectiveness of selected data combinations. The Shapley value seems to be a promising tool for data selection. However, calculating the Shapley value for all the data samples in a dataset is computationally expensive, especially for large-scale fine-tuning datasets.

Motivated by the aforementioned challenges, we present SHED, a Shapley-based automated dataset refinement framework for fine-tuning LLMs. The key intuition behind SHED is to perform Shapley value evaluations on a small portion of representative samples only, thereby dramatically decreasing the computational complexity of Shapley-based data refinement.

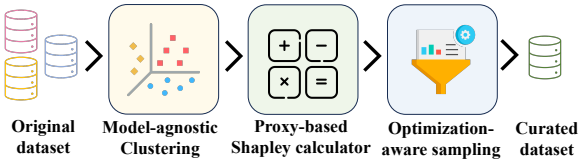


Figure 1: Overview of SHED.

Specifically, as Figure 1 illustrates, SHED consists of three key components: *(1) model-agnostic clustering*, *(2) proxy-based Shapley calculator*, and *(3) optimization-aware sampling*. Initially, the model-agnostic clustering groups embeddings of the original dataset and then selects representative data samples as a proxy for each cluster based on the distance of embeddings to the cluster centroid. These proxy data instances are then evaluated by the proxy-based Shapley calculator, which employs an approximation method to efficiently calculate their Shapley values, focusing on task-specific objectives (*e.g.,* accuracy and fairness). This method involves iteratively removing groups of instances from the proxy dataset and assessing the performance variation of the model to estimate the collective contribution of these instances, thereby streamlining the computation of Shapley values. The derived Shapley values of these proxy data instances are used as the quality score for their respective clusters. Finally, optimization-aware sampling selects data from clusters to compile a compact yet high-quality dataset, employing strategies that may favor clusters with higher-quality scores.

SHED only computes Shapley values for the cluster representatives rather than each data point, drastically boosting the efficiency of data refinement. Furthermore, Yang et al. (2022) observed that hyperparameters tuned on smaller models can be effectively transferred to larger models, significantly reducing tuning costs while maintaining performance [21]. We observed a similar phenomenon: datasets curated by SHED exhibit strong transferability, performing robustly across LLMs of various sizes and families. This suggests that smaller LLMs can be used for data selection, reducing computational costs. The selected datasets can be used to fine-tune larger LLMs and reused in multiple tasks to further amortize costs. Moreover, SHED offers a unified yet flexible framework, catering to various user needs by providing multiple options within each component. For example, the optimization objective for Shapley value measurement can be tailored to specific tasks (*e.g.,* fairness). Our key contributions can be summarized as follows:

- We present SHED, a generic data refinement framework based on Shapley values, which can curate a small yet high-quality dataset for boosting the efficiency of fine-tuning LLMs.
- We conducted extensive experiments on two benchmark datasets, *i.e.,* MMLU and WizzardLM, the results demonstrate that fine-tuning LLMs with small datasets curated by SHED yields performance comparable to, or even better than, using the original large datasets. Notably, datasets curated by SHED exhibit strong transferability, achieving robust performance across various LLMs of different sizes and families. This indicates that smaller models can employed to greatly lower computational expenses for data selection, and the

selected dataset can be used to fine-tune larger models and reused across multiple tasks to further distribute the costs.
- Code associated with the collection of high-quality datasets curated by SHED can be found at SHED: Shapley-Based Automated Dataset Refinement.

2 Related Work

2.1 Coreset Selection

Coreset selection plays a critical role in machine learning by targeting the selection of a representative subset from a larger dataset. Various coreset selection methods use unique criteria for choosing samples. Geometry-based approaches focus on the geometric properties of the data points, striving to retain geometrically significant samples that represent the overall data distribution [22, 23, 24, 25]. Uncertainty-based methods choose samples based on the uncertainty they present to the model, typically engaging samples that the model finds challenging to classify [26, 27, 28]. Decision-boundary-based methods select samples that are close to the decision boundary of the classifier, ensuring that the nuances of the classification boundary are well-represented in the selected subset [29, 30]. Gradient-matching approaches involve selecting a subset that yields similar gradient distributions as the entire dataset when used in training [31, 32]. Bilevel Optimization optimizes the coreset selection in a way that the selected subset maximizes certain performance metrics [33]. Dataset Selection with Datamodels using datamodels to approximate how the learning algorithm utilizes different subsets of training data to minimize target task loss.[34, 35] Submodularity-based approaches consider both diversity and information richness, striving for a balanced representation of the dataset [36].

2.2 Data Selection for Instruction Fine-tuning

Due to the superiority of instruction fine-tuning in enhancing the performance of LLMs, many recent studies focus on selecting high-quality instruction fine-tuning data. Based on methods, it can be divided into the following categories. Indicators-based methods define multiple metrics, such as instruction length and perplexity, to compute quality scores for each instruction instance [16, 37, 38, 39]. Training-based methods leverage the performance improvement through fine-tuning to score and select instruction data suited for fine-tuning [18, 40, 41, 42, 43, 44, 45]. Some other methods employ commercial LLMs like ChatGPT to assess quality, complexity, and diversity of instructions for selection [13, 46, 47, 48, 49].

2.3 Limitations of Previous Work

Most existing methods for data selection overlook the impact of various data combinations on model performance. As Table 1 illustrates, datasets formed by combining high-quality data, which are merely based on the independent quality score of each individual data sample, do not necessarily enhance model performance effectively. The combination of different data can impact the final performance of fine-tuning.

Although TS-DSHAPLEY [18] also utilized Shapley value for data selection, SHED offers several distinct advantages. SHED computes Shapley values only for proxy data of clusters rather than each individual data point, dramatically reducing computational overhead compared to TS-DSHAPLEY. SHED employs model-agnostic clustering, enhancing the transferability of curated datasets across different language models and model families. Moreover, SHED considers data diversity and can be customized for various optimization objectives, while TS-DSHAPLEY primarily focuses on predictive accuracy.

Many other existing works are also task-specific, limiting their applicability. In contrast, SHED offers a unified and flexible framework, adaptable to various instructional tuning tasks, making it more widely applicable.

3 Proposed Method

Motivated by the aforementioned challenges, we present SHED, a generic framework that exploits Shapley value to identify and select high-quality data to improve the performance and efficiency of fine-tuning LLMs.

3.1 Preliminary

The motivation behind this work is underscored by the observation, as illustrated in Table 1, that naively aggregating high-quality data merely based on the independent importance of individual

 https://doi.org/10.52202/079017-3153

Table 1: We apply DSIR [50] to compile a high-quality dataset (10k instances), a random dataset (10k instances) from MMLU, and a mixed dataset samples 5k instances from each of the high-quality and random datasets. We fine-tune the LLaMA-7B model [3] on the curated dataset and evaluate them using the MMLU test set.

Dataset	High-quality	Random	Mixed
MMLU	40.04	39.13	40.92

samples does not guarantee a performance improvement of fine-tuning. We believe this phenomenon is attributed to the complex interactions between different instances within the fine-tuning process. Thus, there is a pressing need to design a novel data selection method, which accounts for the individual and collective contributions of instances to model performance.

The Shapley value offers a compelling solution to this challenge. It quantifies the marginal contribution of each instance to the overall performance of the model, considering all possible combinations of instances. The formulation of the Shapley value for a data sample i in dataset D can be expressed as:

$$S_i = \sum_{P \in D \setminus \{i\}} \frac{|P|!(|D| - |P| - 1)!}{|D|!} (v(P \cup i) - v(P)), \tag{1}$$

where S_i is the Shapley value of i, P is the subset of dataset D, $|D|$ and $|P|$ are the total number of instances in D and P, $v(P)$ is the value function of P, which represents the performance of the LLM model fine-tuned on the subset P. As Eq. 1 indicates, the Shapley value of an instance i captures its average impact on model performance across all subsets it might be part of. This ensures a fair evaluation of the contribution of each instance in the original dataset, enabling the selected data is genuinely beneficial for enhancing model performance when integrated with other data samples.

Additionally, the value function $v(P)$ in Eq. 1 serves to calculate contributions from corresponding data. This value function can be tailored for various optimization objectives, such as accuracy and fairness, facilitating the selection of data that aligns with the task-specific requirements.

However, computing the Shapley value, as depicted in Eq. 1, demands extensive computational efforts, because it requires evaluating the contribution of each instance across all possible combinations. For a dataset with $|D|$ instances, there are a total of $2^{|D|} - 1$ possible combinations. For each combination, two evaluations are needed, *i.e.,* one includes a certain instance and the other one holds out that instance, doubling the computational workload to determine the contribution of that particular instance. Thus, the time complexity for measuring the Shapley value of each instance is $O(2^{|D|})$. Given the need to perform this calculation for all $|D|$ instances to determine their individual Shapley values, the overall time complexity for the dataset increases to $O(|D| \cdot 2^{|D|})$. This exponential complexity makes direct computation of Shapley values impractical for large datasets.

3.2 Design of SHED

To address the above challenges, we design SHED, comprising of three key components: model-agnostic clustering, proxy-based Shapley calculator, and optimization-aware sampling. We introduce each component in detail.

Model-agnostic Clustering. Given the time complexity of computing the Shapley value, calculating the Shapley value for all instances in a large fine-tuning dataset is impractical. The model-agnostic clustering employs models from Sentence Transformers [51] to generate semantically meaningful embeddings for each sample in the original dataset. These embeddings facilitate the efficient and effective computation of semantic similarities between textual inputs, enabling the grouping of data with similar contexts. Moreover, those model-agnostic embeddings enhance the transferability of the curated dataset, as demonstrated in Table 7. Then, the model-agnostic clustering applies algorithms, such as K-means [52] and Agglomerative Clustering [53], to group the embeddings. It then selects the representative data, which is closest to the cluster centroids in the embedding space, for each cluster. In doing so, we use these representative samples as the proxy of the respective clusters. Subsequently, SHED only calculates the Shapley values of those proxy data, using their Shapley values as the quality scores for their respective clusters. Employing proxy data effectively captures the essence of the diversity and complexity in the dataset. This strategy significantly reduces the computational burden associated with calculating Shapley values across vast datasets.

Proxy-based Shapley Calculator. To further improve efficiency for Shapley value calculations, the proxy-based Shapley calculator employs an approximation method to estimate the Shapley values of the proxy data. This method iteratively removes groups of n instances from the proxy data D_p, followed by an evaluation of the model's performance to assess the impact of these instances. The performance variations before and after the removal of a specific group of instances quantify their collective contribution. Specifically, the contribution of the initial group of n instances, denoted as $c_{(1..n)\in D_p}$, is computed by $c_{(1..n)\in D_p} = v(D_p) - v(D_p \setminus \{1..n\})$. Similarly, the contribution for the subsequent group of n instances is determined by $c_{(n+1..2n)\in D_p} = v(D_p \setminus \{1..n\}) - v(D_p \setminus \{1..2n\})$. This procedure is repeated, progressively removing groups of n instances until the entire proxy data has been visited, which marks the completion of a single iteration. This entire iteration process is then repeated k times to enhance the accuracy of the approximation. After completing k iterations, the Shapley value for a certain instance i of the proxy dataset is approximated using the average of its contributions across all iterations, defined as $S_i \approx \frac{1}{k}\sum_k \frac{c_i(k)}{n}$, where $c_i(k)$ denotes the contribution associated with instance i in the kth iteration.

Optimization-aware Sampling. The Shapley value of each proxy data is assigned as the quality score of the corresponding cluster. Optimization-aware sampling utilizes these quality scores to sample data from these clusters, aiming to curate a small yet high-quality dataset. Optimization-aware Sampling offers two sampling methods: Quality-Ordered Cluster Sampling (QOCS) and Quality-Weighted Cluster Sampling (QWCS). QOCS prioritizes sampling from clusters with the highest quality scores. It selects instances starting from the most high-quality clusters until a predefined target sampling number is reached. QWCS adopts a probabilistic approach to sample instances across all clusters, with the probability of selection from a given cluster weighted by its quality score. This method aims to balance quality with diversity by allowing for the inclusion of instances from a broader array of clusters, thus potentially enriching the dataset with a wider variety of high-quality data points. The probability $\Pr(i)$ of selecting an instance from cluster i is defined in Eq. 2:

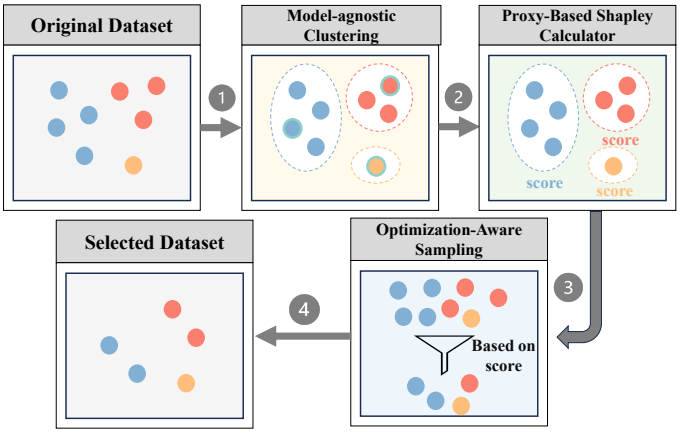


Figure 2: Workflow of SHED: ① Clustering and determining proxy data; ② Calculating Shapley values as scores; ③ Sampling based on scores; and ④ Forming the selected dataset.

$$\Pr(i) = \frac{e^{fS_i}}{\sum_i e^{fS_i}}, \tag{2}$$

where S_i represents the quality score of cluster i, and f is a scaling factor that modulates the emphasis on quality versus diversity within the sampled dataset. By adjusting f, users can tailor the sampling process to prioritize either quality or diversity to suit specific task goals. A higher f value tends towards selecting higher-quality instances, offering a versatile toolkit for dataset optimization.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on two famous benchmark datasets, MMLU (99.8k instances) [54] and WizardLM-evol-instruct-70k (70k instances) [55].

SHED Implementation. We use the K-means algorithm for the model-agnostic clustering and set the number of clusters to 3000. For the proxy-based Shapley calculator, the value function is set as the accuracy of the foundation model fine-tuned on the proxy data. We use LLaMA-7B [3] as the

https://doi.org/10.52202/079017-3153

pre-trained foundation model and 10% instances in the MMLU test set calculating the Shapley values of proxy data. The number of iterations k is set to 10, and the number of instances n removed from the proxy data each step is set to 60. To conserve time and resources, instruction fine-tuning within the proxy-based Shapley calculator is conducted for one epoch. For optimization-aware sampling, we employ the QOCS and QWCS strategies with setting the scaling factor to 1, investigating their efficacy with a variety of target sampling sizes. These implementations are denoted as **SHED-QOCS** and **SHED-QWCS**. The target sampling size varies from $1,000$ to $20,000$ with increments of $1,000$, to thoroughly assess the impact of each sampling approach on fine-tuning performance.

Baseline Methods. We compare SHED with three baseline methods. Specifically, we implement a random-sampling method, denoted as **RS**, which randomly selects a subset from a large dataset. We also use the Dataset Quantization method [38], denoted by **DQ**, and the Data Selection with Importance Resampling [50], denoted by **DSIR**, for comparisons. In addition, we also consider fine-tuning models on the entire dataset, denoted as **FULL**, as a baseline.

Evaluation Settings. After obtaining the curated datasets using SHED and baseline methods, we fine-tune the pre-trained models using each curated subset, respectively. We apply the Low-Rank Adaptation (LoRA), which is a flexible and efficient tool, for fine-tuning and set the default LoRA rank to 128 [56, 57]. For all curated datasets, the instruction fine-tuning was conducted for 3 epochs. Notably, we use the same hyperparameters in fine-tuning across all methods to ensure a fair comparison, aiming to isolate the impact of data selection on model performance. We evaluate the performance of fine-tuned models on MMLU and ARC-challenge tasks using the `lm-evaluation-harness` testing framework [58]. To better evaluate the human preferences of fine-tuned models, we adopt `MT-Bench` [59] in our experiments. All the experiments are conducted on two A100 GPUs, each with 80GB of memory.

4.2 Experiment Results

We summarize the experimental results for SHED and other baseline methods. For consistency, the **bold** numbers indicate the corresponding method outperforms the **FULL** method. Additionally, we underline the best result achieved among all the methods that curate subsets.

For each method, the dataset from the curated collections that yields the optimal result across various sample sizes is referred to as the **best-selected dataset**.

Table 2: Performance comparison of curated datasets of the same size by SHED and baseline methods.

Original dataset	MMLU					WizardLM				
Method	RS	DQ	DSIR	QOCS	QWCS	RS	DQ	DSIR	QOCS	QWCS
MMLU	38.94	39.88	40.24	44.80	43.87	33.12	**33.20**	**33.86**	**35.43**	**34.91**
ARC-challenge	45.10	**46.35**	45.67	**47.10**	**47.23**	46.01	**48.71**	47.66	**49.47**	**49.92**

Table 3: Performance of the best-selected datasets of SHED and baseline methods on the MMLU task.

	MMLU	WizardLM
QOCS	44.80 (10k)	**35.92 (4k)**
QWCS	44.24 (13k)	**35.76 (9k)**
RS	40.87 (15k)	**34.33 (7k)**
DQ	43.50 (7k)	**33.97 (7k)**
DSIR	40.23 (13k)	**34.72 (10k)**
Full	45.56 (99.8k)	33.16 (70k)

Effectiveness of SHED. Given the datasets generated from SHED and the baseline methods, we fine-tune the LLaMA-7B model, respectively, and evaluate the fine-tuned models on the MMLU and ARC-challenge tasks. We compare the results of the datasets of 10k instances curated by SHED and the baseline methods. As depicted in Table 2, when the number of total sampling instances is fixed (10k), the datasets curated by SHED consistently outperform those chosen by baseline methods. We also compare the performance of fine-tuned models using the best-selected dataset by each method. Table 3 shows the evaluation results for the MMLU task. Our method, **SHED-QOCS**, demonstrated superior performance on the MMLU dataset compared to baseline methods, achieving the highest results among the curated datasets. Furthermore, **SHED-QOCS** also led in performance

Table 4: Performance of the best-selected datasets of SHED and baselines on the ARC-challenge task.

	MMLU	WizardLM
QOCS	**47.10 (10k)**	**51.36 (1k)**
QWCS	**49.21 (9k)**	**50.26 (7k)**
RS	**47.07 (13k)**	**49.33 (16k)**
DQ	**46.50 (3k)**	**50.24 (5k)**
DSIR	**46.90 (3k)**	**48.78 (12k)**
Full	45.99 (99.8k)	47.95 (70k)

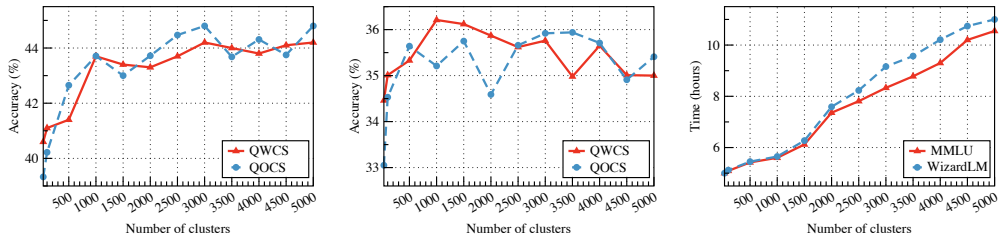
Table 5: MT-Bench evaluation of the best-selected datasets of SHED and baselines.

Original dataset	MMLU					WizardLM				
Method	Full	RS	QOCS	RS	QWCS	Full	RS	QOCS	RS	QWCS
Size	99.8k	10k	10k	13k	13k	70k	4k	4k	9k	9k
LLaMA-7B	3.02	2.23	2.53	2.44	2.83	5.21	4.77	4.89	4.81	**5.24**

when utilizing the WizardLM dataset. It is notable that **SHED-QOCS** outperforms the full dataset, achieving a 2.76% higher accuracy. In Table 4, we report the results of the ARC-challenge task. Similarly, among the datasets curated from the MMLU dataset, the selected dataset of our method **SHED-QWCS** achieves the best result compared with the baseline methods. It also surpasses the full dataset by 3.22%. Within the datasets derived from WizardLM, **SHED-QOCS** once again curated the dataset of best performance, which surpasses the full dataset by 3.41%. The results demonstrate the effectiveness of SHED. Although SHED demands more computational effort, its strength lies in creating high-performance datasets.

Evaluations on MT-Bench. We use MT-Bench to evaluate the performance of datasets curated by SHED in terms of human preferences. Table 5 demonstrates that the datasets curated by SHED align well with human preferences, not only enhancing accuracy but also enabling the model to better understand and follow human instructions, generating answers that are more favorable to humans. The dataset constructed through the SHED-QWCS method, sampled from WizardLM, achieved a remarkable score of 5.24 on the MT-Bench. The results presented in Table 5 represent the average of five independent runs.

Transferability Evaluation of Curated Datasets across Various Models. To evaluate the transferability of datasets curated by SHED, we first apply SHED to select data from the MMLU and WizardLM datasets based on LLaMA-7B. Then, we fine-tune LLaMA-13B, Vicuna-7B, and GPT-2 using the best-selected dataset curated by SHED and the baseline methods. As summarized in Table 6 and Table 7, datasets curated by SHED exhibit robust performance across various models, demonstrating their transferability and applicability across various tasks and even different model families. The strong transferability of the curated datasets indicates that SHED identifies generally high-quality data. The computational cost for data selection can be significantly amortized across various models. In addition, the datasets selected by LLaMA-7B also achieve good performance when fine-tuning the larger model LLaMA-13B. This indicates that we can utilize smaller models to select data, thereby significantly reducing the computational cost of data selection.



(a) Subsets selected from MMLU. (b) Subsets selected from WizardLM. (c) Computational time for one iteration of Shapley value calculation.

Figure 3: Performance of subsets with varying numbers of clusters in SHED.

https://doi.org/10.52202/079017-3153

Table 6: Transferability evaluation using the best-selected datasets across different models on MMLU task.

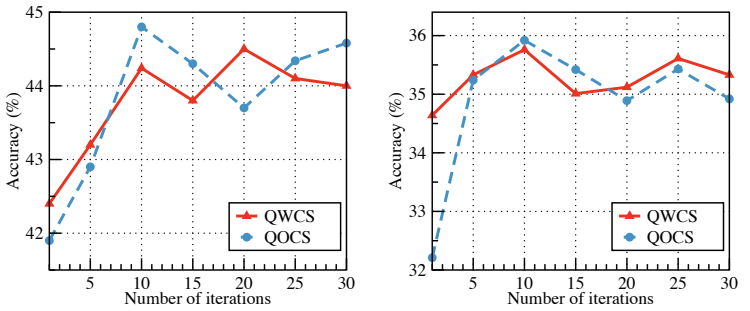
Original dataset	MMLU					WizardLM				
Method	Full	RS	QOCS	RS	QWCS	Full	RS	QOCS	RS	QWCS
Size	99.8k	10k	10k	13k	13k	70k	4k	4k	9k	9k
LLaMA-13B	53.22	50.04	52.95	50.12	51.54	45.63	45.77	**45.93**	45.81	**46.36**
VICUNA-7B	49.70	48.43	**50.01**	47.21	48.93	45.56	45.71	**47.19**	45.44	**48.16**
GPT-2	24.22	23.74	**26.89**	**24.33**	**25.83**	26.19	25.07	**26.76**	24.85	25.77

Table 7: Transferability evaluation using the best-selected datasets across different models on ARC-challenge task.

Original dataset	MMLU					WizardLM				
Method	Full	RS	QOCS	RS	QWCS	Full	RS	QOCS	RS	QWCS
Size	99.8k	10k	10k	13k	13k	70k	4k	4k	9k	9k
LLaMA-13B	49.31	47.31	**50.43**	48.83	**50.68**	54.09	53.17	**55.20**	54.11	**55.63**
VICUNA-7B	44.88	44.86	**45.23**	43.24	**44.91**	49.91	47.72	**50.26**	47.98	48.72
GPT-2	19.45	18.77	**19.81**	19.02	**20.05**	19.19	17.98	**19.28**	18.72	**19.54**

Impact of Number of Clusters. The number of clusters in K-means affects the computational cost needed for Shapley value calculations and the relevance of proxy data to its cluster. An increase in the number of clusters leads to smaller and more homogeneous groups, thereby improving the proxy data's representativeness for its respective clusters. However, this comes at the cost of increased computational overhead, highlighting a balance that must be struck to optimize both efficiency and representativeness. In this experiment, we evaluate the best-selected dataset by SHED across varying numbers of clusters using LLaMA-7B on the MMLU test set. Guided by the findings in [60], our investigation begins with a baseline cluster count of $C = \sqrt{|D|}$. We present the computation time for Shapley value computations across different settings, maintaining consistency with the experimental setup outlined in Section 4.1.

As Figure 3(a) and Figure 3(b) show, the results reveal that performance improvements of curated dataset reach a plateau when the number of clusters exceeds $3\sqrt{|D|}$. Meanwhile, Figure 3(c) demonstrates a proportional increase in computation time for Shapley value calculations as the number of clusters rises. Notably, at very low cluster counts (e.g., below 1000), Shapley value computation times are largely dictated by the evaluation, with the time spent remaining relatively constant across varying datasets. In such cases, the computation time is more significantly affected by the size of pre-trained models rather than the number of clusters itself. Given the transferability of datasets curated using the SHED, it is feasible to employ a smaller foundational model than the target model within the proxy-based Shapley value calculator. In doing so, the computation overhead for evaluation can be significantly reduced, making SHED a practical approach in real-world settings.



(a) Subsets selected from MMLU. (b) Subsets selected from WizardLM.

Figure 4: Performance of subsets with varying iterations in SHED.

Impact of Number of Iterations on Proxy-based Shapley Calculator. The precision of Shapley value estimates increases with the number of iterations k, providing a more accurate measurement of each data sample's contribution to the model performance. However, this increment also leads to a proportional rise in computational cost, leading to a contrasting relationship between computational efficiency and the accuracy of Shapley value estimations. To seek the optimal number of iterations

for Shapley value calculations, we analyzed the performance of datasets curated by SHED under varying iteration settings. The experiments are conducted with the LLaMA-7B model on the MMLU test set, following the experimental settings detailed in Section 4.1.

Figures 4(a) and 4(b) illustrate that the performance of the curated datasets by **QOCS** and **QWCS** are stable once the iteration number surpasses 10. This result highlights the stability of our methods beyond 10 iterations, showing that further iterations beyond this threshold do not significantly improve dataset quality. Given the balance between computational cost and performance, setting the number of iterations to 10 is recommended for optimal efficiency and robustness.

5 Discussion

5.1 Data Selection for Multiple Tasks.

In our experiments, we thoroughly evaluate methods regarding accuracy. It is notable that our framework is readily adaptable. By setting different value functions $v(P)$, SHED can select any subset using arbitrary criteria. This adaptability allows SHED to customize its data selection process to produce a small dataset while improving specific objectives, such as model fairness [61].

In particular, if we aim to curate a dataset using the common fairness notion, *i.e.,* demographic parity, we can define $v(P)$ the disparity in positive prediction rates between groups with protected attributes (e.g., males vs. females), calculated as the negative absolute difference $-|X_{\text{Male}} - X_{\text{Female}}|$, where X_{Male} and X_{Female} are the positive prediction rates for male and female groups, respectively.

5.2 Complexity Analysis

We assume that the running time required to fine-tune the model using a single instance is denoted by t, and the time needed to evaluate the model on a test set consisting of m instances is represented by T_m. Let C denote the number of clusters, n denote the number of instances within a group and k signifies the number of iterations utilized in the proxy-based Shapley calculator as illustrated in Section 3.2. The total number of evaluations and fine-tuning per iteration would be proportional to $\frac{C}{n}$. For simplicity, we assume that C is evenly divisible by n for simplicity. Given k iterations, the overall time complexity of this approximation method can be expressed as $\mathcal{O}\Big(\frac{Ck}{n}\Big[\frac{(C+n)t}{2} + T_m\Big]\Big)$.

6 Conclusion

In this work, we introduced SHED, an innovative Shapley value-based framework designed to refine datasets for the efficient fine-tuning of LLMs, addressing the computational hurdles commonly associated with Shapley value calculations through a novel clustering and proxy-based approach. Through extensive experiments conducted on benchmark datasets such as MMLU and WizardLLM, we have shown that LLMs fine-tuned with datasets curated by SHED not only match but, in some cases, surpass the performance of those trained with the original, larger datasets. Significantly, SHED-curated datasets have demonstrated a high degree of transferability, maintaining robust performance across various models. Furthermore, SHED's flexibility and efficiency underscore its potential to revolutionize LLM fine-tuning by allowing for the creation of compact, high-quality datasets.

7 Limitations

This research, while presenting significant advancements, encounters certain limitations that merit attention for future work. Firstly, the method's reliance on sufficiently representative embeddings may limit its applicability in real-world scenarios where such embeddings are unavailable or inadequate. Future work will explore ways to reduce this dependency for broader applicability. Secondly, the use of clustering and proxy data may overlook rare but important samples. Future research will focus on improving clustering methods to better capture these samples. Additionally, the framework's current objective focuses predominantly on model performance, which may inadvertently lead to model bias. This singular focus overlooks the equally important aspect of model fairness, crucial for ensuring that models perform equitably across diverse groups. Recognizing this, our framework is designed to be extensible and objective-agnostic, laying the groundwork for incorporating additional criteria. In subsequent research, we plan to integrate considerations of model fairness alongside performance.

8 Ethics Statement

In this work, we present SHED, a generic data refinement framework utilizing Shapley values, aimed at assembling a compact yet effective dataset to boost the efficiency of the fine-tuning process of

 https://doi.org/10.52202/079017-3153

LLMs. This study carefully avoids ethical issues beyond standard AI concerns, leveraging properly cited publicly available Internet text data. This approach ensures adherence to ethical data use standards, reflecting our commitment to responsible research practices in the AI field.

Acknowledgements

We thank the anonymous reviewers for their valuable insights and recommendations, which have greatly improved our work. This research has been graciously funded by NSF 2431611.

References

[1] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[4] 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.

[5] Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, et al. Llm360: Towards fully transparent open-source llms. *arXiv preprint arXiv:2312.06550*, 2023.

[6] 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.

[7] Ilker Yildirim and LA Paul. From task structures to world models: what do llms know? *Trends in Cognitive Sciences*, 2024.

[8] Ping Guo, Fei Liu, Xi Lin, Qingchuan Zhao, and Qingfu Zhang. L-autoda: Large language models for automatically evolving decision-based adversarial attacks. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '24 Companion, page 1846–1854. ACM, July 2024.

[9] Dun Zeng, Yong Dai, Pengyu Cheng, Longyue Wang, Tianhao Hu, Wanshun Chen, Nan Du, and Zenglin Xu. On diversified preferences of large language model alignment, 2024.

[10] Ankit Srivastava, Piyush Makhija, and Anuj Gupta. Noisy text data: Achilles' heel of bert. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 16–21, 2020.

[11] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023.

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

[13] Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. Alpagasus: Training a better alpaca with fewer data, 2023.

[14] Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Junhao Liu, Tongliang Liu, Fei Huang, and Yongbin Li. One shot learning as instruction data prospector for large language models, 2024.

[15] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp, 2019.

[16] Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. Instruction mining: When data mining meets large language model finetuning, 2023.

[17] Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics, 2020.

[18] Stephanie Schoch, Ritwick Mishra, and Yangfeng Ji. Data selection for fine-tuning large language models using transferred shapley values, 2023.

[19] Alvin E Roth. *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press, 1988.

[20] Benedek Rozemberczki, Lauren Watson, Péter Bayer, Hao-Tsung Yang, Olivér Kiss, Sebastian Nilsson, and Rik Sarkar. The shapley value in machine learning, 2022.

[21] Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer, 2022.

[22] Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding, 2012.

[23] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach, 2018.

[24] Samarth Sinha, Han Zhang, Anirudh Goyal, Yoshua Bengio, Hugo Larochelle, and Augustus Odena. Small-gan: Speeding up gan training using core-sets. In *International Conference on Machine Learning*, pages 9005–9015. PMLR, 2020.

[25] Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 137–153. Springer, 2020.

[26] Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. *arXiv preprint arXiv:1906.11829*, 2019.

[27] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

[28] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34:20596–20607, 2021.

[29] Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. *arXiv preprint arXiv:2109.03764*, 2021.

[30] Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018.

[31] Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pages 5464–5474. PMLR, 2021.

[32] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR, 2020.

 https://doi.org/10.52202/079017-3153

[33] Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8110–8118, 2021.

[34] Logan Engstrom, Axel Feldmann, and Aleksander Madry. Dsdm: Model-aware dataset selection with datamodels, 2024.

[35] Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data, 2022.

[36] Rishabh Iyer, Ninad Khargoankar, Jeff Bilmes, and Himanshu Asanani. Submodular combinatorial information measures with applications in machine learning. In *Algorithmic Learning Theory*, pages 722–754. PMLR, 2021.

[37] Lai Wei, Zihao Jiang, Weiran Huang, and Lichao Sun. Instructiongpt-4: A 200-instruction paradigm for fine-tuning minigpt-4. *arXiv preprint arXiv:2308.12067*, 2023.

[38] Daquan Zhou, Kai Wang, Jianyang Gu, Xiangyu Peng, Dongze Lian, Yifan Zhang, Yang You, and Jiashi Feng. Dataset quantization, 2023.

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

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

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

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

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

[44] Yongrui Chen, Haiyun Jiang, Xinting Huang, Shuming Shi, and Guilin Qi. Tegit: Generating high-quality instruction-tuning data with text-grounded task design. *arXiv preprint arXiv:2309.05447*, 2023.

[45] Po-Nien Kung, Fan Yin, Di Wu, Kai-Wei Chang, and Nanyun Peng. Active instruction tuning: Improving cross-task generalization by training on prompt sensitive tasks. *arXiv preprint arXiv:2311.00288*, 2023.

[46] Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. # instag: Instruction tagging for analyzing supervised fine-tuning of large language models. *arXiv e-prints*, pages arXiv–2308, 2023.

[47] Yang Xu, Yongqiang Yao, Yufan Huang, Mengnan Qi, Maoquan Wang, Bin Gu, and Neel Sundaresan. Rethinking the instruction quality: Lift is what you need, 2023.

[48] 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.

[49] Yingxiu Zhao, Bowen Yu, Binyuan Hui, Haiyang Yu, Fei Huang, Yongbin Li, and Nevin L Zhang. A preliminary study of the intrinsic relationship between complexity and alignment. *arXiv preprint arXiv:2308.05696*, 2023.

[50] Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling, 2023.

[51] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[52] Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8):1295, 2020.

[53] Fionn Murtagh and Pierre Legendre. Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? *Journal of classification*, 31:274–295, 2014.

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

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

[56] 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, 2021.

[57] Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations, 2024.

[58] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023.

[59] 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. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

[60] Shi-Bing Zhou, Zhen-Yuan Xu, and Xu-Qing Tang. Method for determining optimal number of clusters in k-means clustering algorithm. *Journal of computer applications*, 30(8):1995, 2010.

[61] Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. Reducing sentiment bias in language models via counterfactual evaluation. *arXiv preprint arXiv:1911.03064*, 2019.

[62] Paul T Boggs and Jon W Tolle. Sequential quadratic programming. *Acta numerica*, 4:1–51, 1995.

 https://doi.org/10.52202/079017-3153

Appendix

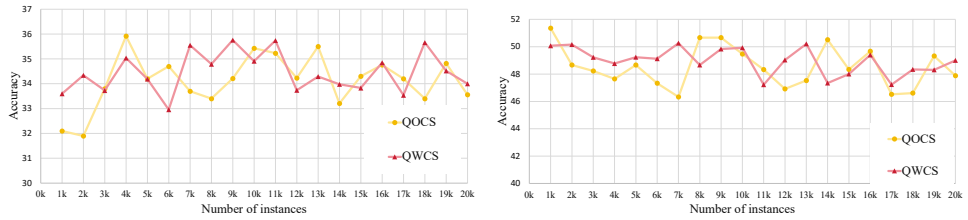
A Hyperparameter Settings and Experimental Configuration

In our experiments, we employed the following hyperparameters: the number of training epochs was set to 3, the batch size was 128, the LoRA rank (`lora_r`) was 128, and the LoRA alpha (`lora_alpha`) was 256. For clustering, when the number of clusters (C) was 3000, the number of samples removed per group (n) was 60; when testing the impact of different C values on performance, $n = \frac{C}{50}$. The number of iterations for Shapley value calculation (k) was 10, and the learning rate was 3×10^{-4}. Data preprocessing involved using the `sentence-transformers/all-MiniLM-L6-v2` model to generate semantically meaningful embeddings for each sample in the original dataset, followed by applying the k-means algorithm to cluster these embeddings.

B Comparison of the SHED-QOCS and SHED-QWCS

We compared the performance of datasets of varying sample sizes using SHED-QOCS and SHED-QWCS methods. For fine-tuning, we utilized the LLaMA-7B model. Other experimental configurations were aligned with the parameters detailed in Section 4.1.

(a) Original dataset: MMLU. Test set: MMLU (b) Original dataset: MMLU. Test set: ARC-challenge



(c) Original dataset: WizardLM. Test set: MMLU (d) Original dataset: WizardLM. Test set: ARC-challenge

Figure 5: Results of curated datasets of different samples.

As figure 5 shows, datasets sampled using the SHED-QWCS approach generally outperform those obtained through SHED-QOCS, particularly with smaller sample sizes. This discrepancy is likely attributable to the limitation of SHED-QOCS in scenarios where the sample size is minimal. In such cases, SHED-QOCS tends to sample data from a limited number of clusters, leading to significant redundancy in the curated dataset.

Conversely, it is observed that datasets that achieve the best performance are often those sampled via SHED-QOCS. This improved performance is observed when the sample size is sufficiently large, allowing SHED-QOCS to sample data from a wider range of clusters. The inherent strength of SHED-QOCS lies in its strategic focus on harvesting high-quality data. As the sample size increases to a point where SHED-QOCS can effectively draw from multiple clusters, its advantage in prioritizing data quality becomes significantly beneficial.

Therefore, we suggest that users select between these two sampling methods based on the sample size they desire.

https://doi.org/10.52202/079017-3153

C Hyperparameter Tuning for Shapley Value Computation

To enhance user convenience, SHED introduces a method for setting hyperparameters.

Based on Figure 3(c), the time per iteration when calculating Shapley values exhibits an approximate linear relationship with the number of clusters. Similarly, the total time for computing Shapley values closely aligns linearly with the number of iterations. Thus, the computation time t for Shapley values can be modeled as $t = \theta kC$. SHED randomly samples 2000 instances from the dataset to calculate the Shapley value for one iteration, recording this to determine θ. To optimize k to be close to 10 and the number of clusters near $3\sqrt{|D|}$, an optimization problem is formulated in Eq. 3.

$$\begin{aligned} \min_{k,C} \quad & \lambda_1(k-10)^2 + \lambda_2(C - 3\sqrt{|D|})^2 \\ \text{s.t.} \quad & \theta kC = t_0 \end{aligned} \tag{3}$$

where λ_1 and λ_2 serve as weights, both defaulting to 1, while t_0 represents the maximum runtime set by the user. This optimization problem can be solved using the SQP (Sequential quadratic programming) method [62] to determine the optimal number of clusters and iterations.

D Comparison with Other Methods

We compare SHED with two methods, LIMA and IFD (using WizardLM) [12, 40]. LLaMA-7B is used as the base model. The results are presented below, focusing on performance in the MMLU and ARC-challenge tasks.

The following table shows the performance of SHED's selected dataset (with 1k samples) compared to LIMA's selected dataset (with 1k samples). It is important to note that SHED is fully automated, whereas LIMA involves manual curation.

Task	**SHED (1k)**	**LIMA (1k)**
MMLU	41.71	34.9
ARC-challenge	48.12	46.16

Table 8: Performance comparison between SHED and LIMA.

We compare SHED with IFD (with 7k samples) to select data from WizardLM. The following table shows the results.

Task	**SHED (1k)**	**SHED (7k)**	**IFD (7K)**
MMLU	33.62	35.63	33.08
ARC-challenge	51.36	50.11	52.90

Table 9: Performance comparison between SHED and Cherry-LLM (using WizardLM).

As the results demonstrate, SHED achieves competitive performance across both tasks, even with fewer samples.

https://doi.org/10.52202/079017-3153

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the development of SHED, an automated dataset refinement framework based on Shapley value for instruction fine-tuning, which aligns with the contributions and scope discussed in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper has a dedicated "Limitations" section where it discusses the reliance on representative embeddings, which may reduce applicability in scenarios lacking such embeddings. It also discusses the potential oversight of rare samples in clustering and the risk of model bias due to a primary focus on performance.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not focus on theoretical results but rather on the implementation and empirical evaluation of SHED.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The experimental setup, including datasets, baseline methods, and evaluation settings, is described in detail in the paper (Section 4 Experiments and Appendix A), making it possible to reproduce the main results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

https://doi.org/10.52202/079017-3153

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have submitted the code and data as supplemental material. All code associated with the collection of high-quality datasets curated by SHED is open-sourced.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have submitted the code and data as supplemental material. All code associated with the collection of high-quality datasets curated by SHED is open-sourced. The details can also be found in the Experiments section and Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports experimental results as averages over multiple runs, which provides some level of statistical significance. While error bars or confidence intervals are not explicitly mentioned, the use of multiple independent runs helps to demonstrate the reliability and consistency of the results.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper details the computational resources used, mentioning the experiments were conducted on two A100 GPUs, each with 80GB of memory, which provides clarity on the computational requirements. Additionally, the paper includes experiments to estimate the runtime under different hyperparameter settings, providing insights into the computational efficiency and resource requirements of the proposed method.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper includes an Ethics Statement section where it discusses the careful avoidance of ethical issues beyond standard AI concerns, ensuring adherence to ethical data use standards and reflecting the commitment to responsible research practices in the AI field.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

https://doi.org/10.52202/079017-3153

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses the broader impact of SHED, emphasizing its potential to significantly enhance the efficiency of fine-tuning large language models while reducing computational costs. It also highlights the transferability of the curated datasets across various models. Additionally, the paper addresses potential negative impacts, such as the risk of model bias due to data selection methods and the implications for model fairness.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: The paper does not explicitly discuss safeguards to prevent misuse of the proposed method.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper uses publicly available benchmark datasets (MMLU and WizardLM) and cites the original papers that introduced these datasets. The licenses and terms of use for these datasets are respected, and the paper provides references to the original sources.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper introduces curated datasets using the SHED framework. All code and datasets are open-sourced, including proper documentation to ensure that other researchers can understand and utilize the assets effectively. The paper provides detailed information about the datasets, the methodology used for their creation, and the associated code.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

https://doi.org/10.52202/079017-3153

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects, and therefore, IRB approval or equivalent is not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.