Balanced Data Sampling for Language Model Training with Clustering

Anonymous ACL submission

Abstract

Data plays a fundamental role in the training of Large Language Models (LLMs). While attention has been paid to the collection and composition of datasets, determining the data sampling strategy in training remains an open question. Most LLMs are trained with a simple strategy, random sampling. However, this sampling strategy ignores the unbalanced nature of training data distribution, which can be sub-optimal. In this paper, we propose ClusterClip Sampling to balance the text distribution of training data for better model training. Specifically, ClusterClip Sampling utilizes data clustering to reflect the data distribution of the training set and balances the common samples and rare samples during training based on the cluster results. A repetition clip operation is introduced to mitigate the overfitting issue led by samples from certain clusters. Extensive experiments validate the effectiveness of ClusterClip Sampling, which outperforms random sampling and other cluster-based sampling variants under various training datasets and large language models.

1 Introduction

Large Language Models (LLMs) have opened new frontiers in understanding and generating human languages (Brown et al., 2020; Touvron et al., 2023a; OpenAI, 2023). A critical aspect of training these models lies in the acquisition and organization of training data (Wang et al., 2023). Some works focus on selecting high-quality data. These works usually perform data filtering or data cleaning from a large corpus, based on either rule-based (Gao et al., 2021; Computer, 2023) or model-based algorithms (Abbas et al., 2023; Tirumala et al., 2023; Marion et al., 2023). On the other hand, several approaches concentrate on optimizing the composition weights of the collected data. These methods typically focus on optimizing the domain weights either using heuristics (Du et al., 2022; Touvron et al., 2023a) or model statistics (Xie et al., 2023a; Fan et al., 2023). While much attention has been given to the collection and composition of diverse datasets for training LLMs (Gao et al., 2021; Computer, 2023; Touvron et al., 2023b; Mukherjee et al., 2023), it is still unclear how the training data sampling affects the optimization of language models.

The sampling methods of existing works are coarse-grained, which determines the sampling weights or sampling order of each domain. The *domain* is usually defined based on the data source or other metadata during the dataset collection, which is coarse-grained and inaccurate. For instance, the Llama models (Touvron et al., 2023a) assign domain mixture weights heuristically, including 67% CommonCrawl web data, 4.5% Github code, 4.5% Wikipedia documents, etc. And Rozière et al. (2023); Azerbayev et al. (2023) improve certain abilities of LLMs by tuning the model on specific domains, like code or mathematical texts. However, the texts are sampled randomly in each domain, which ignores the unbalanced distribution of the expressed meanings and topics. Due to the nature of the data corpus, texts with similar meanings have a long tail distribution in the training set (Zipf, 1949; Chan et al., 2022; Abbas et al., 2023). When using random sampling, LLMs can underfit rare documents and overfit common samples. It is straightforward to utilize a uniform sampling strategy that samples texts with different meanings evenly, which up-samples rare documents and down-samples common texts. However, uniform sampling will up-weight rare documents and repeat them so many times in the training, resulting in severe overfitting of trained LLMs on these training samples.

To address the above challenges, we propose **ClusterClip**, a cluster-based sampling strategy with clip operation to mitigate overfitting. This sampling strategy has two steps. Firstly, we leverage data clustering to reflect the data distribution.

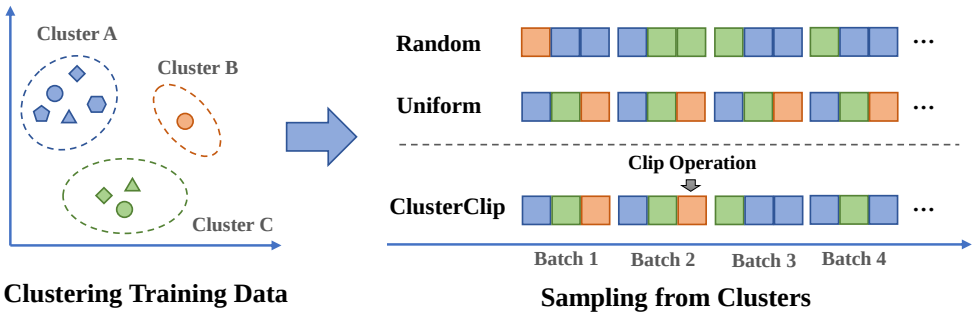


Figure 1: Illustration of **ClusterClip Sampling**. The algorithm utilizes data clustering to describe the training data distribution. Then, it balances the sampling probabilities of samples in different clusters during the training. Moreover, a clip operation is introduced to knock out samples with too many repetitions.

Based on off-the-shelf NLP tools, the semantic-related texts could be grouped into the same cluster. By calculating the size of each semantic cluster, we can evaluate the data rarity. Secondly, we perform the data sampling using the cluster information. The documents from different clusters are sampled evenly at the beginning of the training, thus encouraging the model to learn from rare documents instead of wasting computation on common texts. As the training progresses, a clip operation is applied. If certain documents are sampled too many times, these documents are clipped and no longer sampled from the dataset. Thus, ClusterClip rebalances the data distribution, which facilitates the model to learn rare documents but avoids severe overfitting of repeated texts by clipping.

Our approach is distinguished by its ability to improve model learning efficiency and generalization without relying on dataset-specific metadata or complicated optimizations. Extensive experiments demonstrate the versatility and effectiveness of our proposed ClusterClip Sampling. We show that it consistently enhances the performance of representative LLMs, Llama2-7B (Touvron et al., 2023b) and Mistral-7B (Jiang et al., 2023), in both pre-training and supervised fine-tuning scenarios, indicating its broad applicability. Moreover, we further evaluate several variants of cluster-based sampling methods and demonstrate that: (1) All the cluster-based sampling method variants outperform random sampling, indicating the effectiveness of sampling based on semantic distribution; (2) The Clip operation effectively mitigates the overfitting of repeated documents, leading to large improvements in diverse tasks. (3) Specializing data training order (e.g. General-to-Specific or Specific-to-General) can affect the performance of LLMs on various downstream tasks, showing the promise of progressive learning on LLMs;

To sum up, our contributions are as follows:

- We propose ClusterClip Sampling, which rebalances the occurrence of common or rare documents with a clip operation to avoid severe overfitting.
- We validate the effectiveness of ClusterClip Sampling on multiple datasets and LLMs, demonstrating the broad applicability and stable improvements across pre-training and fine-tuning.
- We present representative variant cluster-based sampling methods to show where the improvements of ClusterClip come from and provide interesting results for future design of sampling strategies.

2 Related Works

Data Clustering for LLMs Clustering methods are widely applied in data curation of LLM pre-training. A line of work incorporates clustering for data deduplication. Semdedup (Abbas et al., 2023) aims to eliminate semantic duplicates from the pre-training corpus and use clustering in the sentence embedding space to reduce the computational cost. D4 (Tirumala et al., 2023) further aggressively removes duplicate documents by combining multiple clustering-based deduplication methods. Moreover, clustering also has been used to improve data quality. Existing works perform quality filtering using a classifier to find data samples that are close to high-quality texts (Brown et al., 2020; Gao et al., 2021; Du et al., 2022; Touvron et al., 2023a),. And MiniPile (Kaddour, 2023) is constructed by filtering out low-quality clusters based on semantic embeddings.

Different from previous works, we explore the effectiveness of clustering for learning strategies instead of data curation for LLM training.

Data Composition for LLM Pre-Training The composition of pre-training data is a determinant of LLM performance. Efforts are made to collect and utilize domain mixtures of pre-training data to improve LLM performance (Longpre et al., 2023; Shen et al., 2023; Nijkamp et al., 2023). Brown et al. (2020); Chung et al. (2022); Du et al. (2022); Azerbayev et al. (2023); Touvron et al. (2023a,b) exploit manually designed domain composition weights by small-scale pre-training experiments. However, we focus on cluster-based data sampling approaches, which do not rely on manual selection of data composition. In addition, existing works also explore the algorithms for searching optimal domain weights. Several methods involve training auxiliary models, either proxy models or reference models to determine the composition weights (Xie et al., 2023a; Fan et al., 2023; Mindermann et al., 2022). Furthermore, some works concentrate on sample-level data selection by introducing model-based metrics to measure sample weights, which include importance score (Xie et al., 2023b), perplexity (Xia et al., 2023), and gradients (Marion et al., 2023). Different from these works, the cluster-based sampling strategies utilize off-the-shelf embedding models and semantic-based cluster-level data sampling.

3 Methodology

We propose **ClusterClip** Sampling, a cluster-based sampling strategy to rebalance the data distribution of the training corpus to facilitate model learning and mitigate overfitting. In this section, we delve into the detailed process of data clustering and the sampling strategy using cluster information to balance the sampling probabilities of common and rare documents.

3.1 Data Clustering

Aiming to describe and manipulate the distribution of texts in the training set, we introduce data clustering to group samples into semantic clusters. We choose not to rely on metadata from the dataset itself, as such information is often absent or fuzzy (Gao et al., 2021; Azerbayev et al., 2023). Instead, data clustering can automatically discover semantic similar documents and group these data points into clusters. Specifically, we first utilize off-the-shelf transformer-based models to generate text representations for each data sample. Then we conduct a K-Means clustering on these generated data embeddings to group samples into clusters. By clustering, we classify data with similar topics into the same subset. Thus, we can analyze the data distribution and rebalance the data distribution when sampling the training data.

We choose out-of-the-box transformer-based embedding models and the K-Means method in the experiments as these methods are well-established, efficient at scale, and can produce semantic-related data clusters. Other embedding functions, including rule-based or model-based, could also be utilized for more accurate clustering. Comparing the impact of different embedding or clustering methods on the data sampling strategies would be a valuable topic and is our future work.

3.2 ClusterClip Sampling

After data clustering, the clusters can describe the long tail distribution of the training set. Based on the cluster information, ClusterClip Sampling increases the sampling weights of rare documents and decreases the weights of common texts. Moreover, a clip operation is introduced to mitigate overfitting. Thus, ClusterClip Sampling balances the learning on both very common texts and extremely rare documents.

Uniform Sampling At the beginning of training, ClusterClip Sampling performs a Uniform Sampling from the clusters, which aims to up-sample rare data points and down-sample common texts. We ensure that each cluster has the same probability of being sampled. After sampling the cluster, amount of tokens in each cluster. This also improves the data diversity within the batch as it balances the occurrence of samples in each cluster in a batch.

Clip Operation When uniformly sampling the data, documents from small clusters can be sampled a huge number of times. In this case, the model will suffer from overfitting on these small semantic clusters and not learn well on the whole training set. To solve this issue, we further propose a clip operation to add a maximum repetition of each sample. When different clusters are uniformly sampled, small clusters can be consumed multiple times. The ClusterClip will record the repeated times of each cluster. When one cluster has been consumed a certain number of times, the cluster

will be knocked out and will not be sampled in further training. Thus, the model will see a sample at most a certain number of times, which mitigates the overfitting.

4 Experimental Setups

To validate the proposed ClusterClip Sampling, we conduct extensive experiments on multiple datasets and LLMs. In this section, we introduce the experimental setups of our experiments, including the baselines, training datasets, training hyperparameters, and evaluation setups.

4.1 Baselines

To demonstrate the effectiveness of ClusterClip Sampling, we introduce several representative sampling methods for comparison.

Random The texts are sampled randomly, which is widely used in the pre-training and fine-tuning of many LLMs (Touvron et al., 2023a,b; Azerbayev et al., 2023; Mukherjee et al., 2023).

Uniform The texts are uniformly sampled from each cluster, which is a simplified version of ClusterClip without the clip operation.

General-to-Specific (G2S) Inspired by recent practice of training domain-specific language models (Rozière et al., 2023; Azerbayev et al., 2023), we want the model to learn general abilities before acquiring specific domain knowledge and skills. Thus, G2S initiates by uniform sampling from each cluster. If a particular cluster is exhausted, sampling from that cluster ceases until the entire dataset has been traversed. By doing so, the model learns diverse and general samples before concentrating on some specific clusters.

Specific-to-General (S2G) The objective of S2G is to prioritize the model's learning of rare samples and can be viewed as the opposite of G2S. It involves initially training the model to acquire knowledge in some specific domains before learning general capabilities. To achieve this, it employs the exactly reversed sampling order of the G2S strategy for training. This can be related to progressive training or curriculum learning (Bengio et al., 2009; Hacohen and Weinshall, 2019), in which the model first learns from easy samples and then transfers to hard ones.

4.2 Training Datasets

To fully validate the effectiveness of ClusterClip Sampling, we train the models with different sampling methods, including the proposed method and baselines, on both supervised fine-tuning and pre-training setups.

Superivsed Fine-Tuning Dataset We choose **Open-Orca** (Lian et al., 2023) to probe the sampling strategies on supervised fine-tuning. It is an open-source implementation of Orca (Mukherjee et al., 2023) that employs GPT-3.5 and GPT-4 to generate detailed answers and intermediate thoughts given a diverse set of NLP tasks. Following the methodology of Orca, Open-Orca collects 1 million outputs from GPT-4 and 3.2 million outputs from GPT-3.5 based on these inputs. The total size of the training set is about 1B tokens.

Pre-Training Dataset We utilized the **Proof-Pile-2** dataset (Azerbayev et al., 2023) to investigate the influence of various sampling strategies for continual pre-training on specific domains. The Proof-Pile-2 dataset is motivated by enhancing the mathematical reasoning capabilities of models and consists of three components: code files, web pages, and scientific papers, leading to 55B tokens in total.

Preprocessing and Clustering We use the base-sized Jina Embeddings 2[1] (Günther et al., 2023) with mean pooling to generate text embeddings. For Open-Orca, we do not differentiate between data generated by GPT-4 and GPT-3.5 and mix them for clustering and training, which reflects the results in a more realistic scenario with mixed instruction data quality. We concatenate the inputs and outputs and truncate the result text into 1024 tokens for embedding computation. Then, we run K-Means with cosine distance on generated embeddings over 300 iterations to obtain 2000 clusters. For Proof-Pile-2, we combined and shuffled different sub-domains of the mixture together and sampled 10B tokens as our training set. This approach allows us to explore the model behavior when changing data sampling strategies in the context of continual training on data mixtures with multiple domains. After obtain the document embeddings, we set the number of iterations and clusters of K-Means to 300 and 100 to obtain the clusters, respectively.

[1] https://huggingface.co/jinaai/jina-embeddings-v2-base-en

4.3 Training Setups

We adopt representative LLMs for training using the InternLM (Team, 2023). All models are trained on 64 A800 GPUs with bfloat16 mixed precision.

Supervised Fine-Tuning Setups We choose Mistral-7B (Jiang et al., 2023) without aligning for chat as the backbone for supervised-fine-tuning. We fine-tuned the model with these sampling strategies separately, each for 20000 steps with a global batch size of 0.25 million tokens on Open-Orca, totaling 5B tokens. The context length is 4096 with packing. The learning rate is warmed up to $3e-6$ over the first 200 steps and then cosine decayed to $3e-7$ at the end of training. Following Muennighoff et al. (2023), the threshold of clipping is set to 5 for all experiments unless specified. To train one model, it utilizes 384 GPU hours.

Pre-Training Setups We initialize the backbone model from the Llama2-7B (Touvron et al., 2023b) base model for continual pre-training. We trained the model with each sampling method for 5000 steps with a global batch size of 4 million tokens, which used 20B tokens in total. The context length is 4096 with packing, and the learning rate is first warmed up to $5e-5$ over 200 steps and then cosine decayed to $1e-5$ at the end of training. The training utilizes 1216 GPU hours for each model.

4.4 Evaluation Setups

We introduce a diverse set of evaluation tasks and datasets to reflect the general and detailed model performance when incorporating different sampling methods.

Evaluation Datasets The trained models are evaluated on a wide range of downstream tasks, including SuperGLUE (Wang et al., 2019), GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), OpenBookQA (Mihaylov et al., 2018), MMLU (Hendrycks et al., 2021a), BBH (Suzgun et al., 2023) and MT-Bench (Zheng et al., 2023).

Evaluation Methods Following Team (2023), we use perplexity-based evaluation for SuperGLUE, OpenBookQA, and MMLU. The few-shot chain-of-thought prompting (Wei et al., 2022) is used to evaluate the accuracy of reasoning tasks, including GSM8K, MATH and BBH. The MT-Bench is evaluated by GPT-4 with reference-based scoring prompts (Zheng et al., 2023).

5 Experimental Results

In this section, we present the experimental results, including the comparison of the proposed sampling method with baselines (Sec 5.1), a detailed analysis of multiple variants of cluster-based sampling methods (Sec 5.2), and the ablation study of different parts of ClusterClip (Sec 5.3).

5.1 Main Results

Supervised Fine-Tuning Experimental results on Open-Orca show the effectiveness of ClusterClip Sampling on the overall model performance across multiple domains and capabilities. As shown in Table 1, the model trained on Open-Orca with ClusterClip Sampling outperforms Uniform Sampling on SuperGLUE, OpenBookQA, and MT-Bench, and beats the Random Baselines with a large margin. The ClusterClip also achieves comparable performance when compared with the sampling methods G2S and S2G on Open-Orca. It demonstrates that the ClusterClip alleviates the overfitting issue of Uniform Sampling and improves the overall performance.

	SuperGLUE	GSM8K	OpenBookQA	MT-Bench
Mistral-7b	50.19	47.61	64.20	-
Random	62.11	61.49	79.80	6.60
Uniform	63.00	58.83	78.20	6.75
G2S	**65.41**	59.36	79.40	6.81
S2G	64.95	**62.55**	80.20	**7.08**
ClusterClip	64.30	58.68	**81.40**	6.90

Table 1: Comparison of different sampling strategies on Open-Orca.

Continual Pre-Training We also demonstrate the effectiveness of ClusterClip Sampling in continual pre-training. As shown in Table 2, the model trained on Proof-Pile-2 with ClusterClip obtains strong overall performance, which achieves 7.90 on MATH and 51.05 on MMLU. This demonstrates that the ClusterClip Sampling outperforms other sampling methods on Proof-Pile-2. We also notice that the ClusterClip Sampling consistently outperforms Uniform Sampling both on Proof-Pile-2 and Open-Orca, which demonstrates that the overfitting issue is significant for Uniform Sampling and the Cutoff strategy alleviates this issue while still maintaining the benefits of data diversity of Uniform Sampling. Moreover, the G2S and S2G sampling methods underperform the ClusterClip method on all these downstream tasks, which are not consistent compared with results from supervised fine-

tuning. It indicates that the effect of changing the training order is unstable. And ClusterClip Sampling is effective in both continual pre-training and supervised fine-tuning, demonstrating the generalization and board application.

	MATH	GSM8K	MMLU	BBH
LLama2-7B	3.50	16.68	46.79	38.20
Random	6.52	25.55	48.84	41.81
Uniform	7.62	**26.00**	49.98	**42.89**
G2S	6.92	23.43	49.42	41.69
S2G	6.98	23.12	48.61	40.90
ClusterClip	**7.90**	24.79	**51.05**	42.78

Table 2: Comparison of different sampling strategies on Proof-Pile-2.

Results on Different Domains We find that ClusterClip can also boost the general performance of large language models on different domains. As shown in Figure 2, the model trained with ClusterClip improves scores of all subsets of MMLU. The results on MMLU demonstrate that the model trained with ClusterClip can generalize across diverse tasks and domains even though the training set Proof-Pile-2 mainly targets mathematical tasks. Besides, Uniform Sampling also improves the performance when compared with Random Sampling, but still underperforms ClusterClip in all the categories of MMLU. Notably, the special cluster-based sampling methods (G2S and S2G Sampling) do not generalize well across different subsets of MMLU, which even underperform Random Sampling on humanities, stem, or social-science categories.

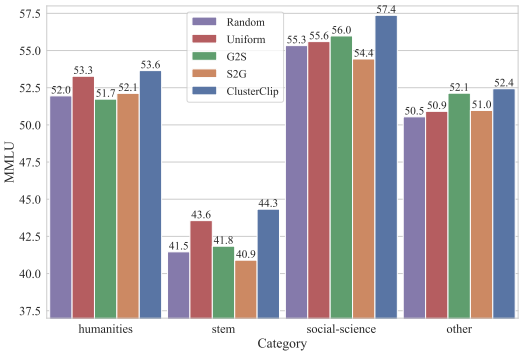


Figure 2: Accuracy of models trained with different sampling methods on each subset of MMLU.

Results on Different Models We also investigate the effectiveness of the proposed methods under different models. We train Llama2-7B on Open-Orca and compare the results with the performance of Mistral-7B. The training setups are kept the same in the supervised fine-tuning setups but the peak learning rate is set to 1e-5 to meet the requirements of fine-tuning Llama2-7B. As shown in Figure 3, ClusterClip consistently outperforms Random and Uniform Sampling by a large margin on both Llama2 and Mistral. However, the other three cluster-based sampling variants only improve marginal performance on different models, compared with Random Sampling.

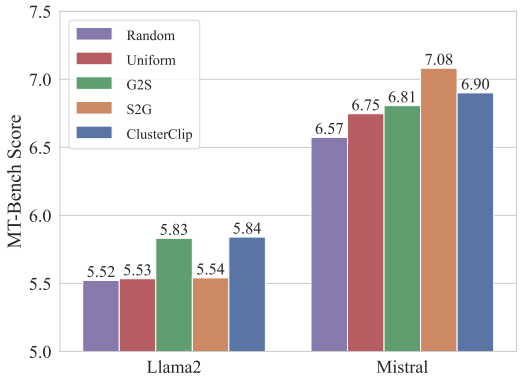


Figure 3: The results of different sampling methods on Llama2-7B and Mistral-7B, measured by MT-Bench Score.

5.2 Analysis ClusterClip Sampling in Training

To find out how the ClusterClip Sampling affect the model performance as the training goes, we massively evaluate the intermediate checkpoints of models trained with different sampling strategies. The supervised fine-tuning results are shown in Figure 4 and the pre-training results are shown in Figure 5. We also present the data distribution of the training set in Figure 6 to see the connection of the data distribution and the results of these sampling methods.

Analysis on Supervised Fine-Tuning As shown in Figure 4, all sampling strategies based on clustering outperform Random sampling, which demonstrates that cluster information can provide insights for sampling strategy design. Moreover, Random sampling improves the instruction-following ability at first but is quickly saturated and even overfitting at the end of the training. This indicates that Random sampling is unstable and sub-optimal for Open-Orca fine-tuning. Comparing clustering-based sampling methods, we surprisingly find that all these methods consistently improve the MT-

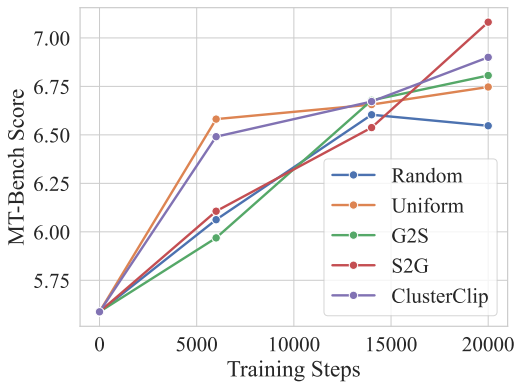


Figure 4: MT-Bench Scores of different sampling strategies during the training on Open-Orca dataset.

Bench score across the training phases. Uniform sampling achieves good performance in the early stage but improves slowly, which may result from cluster repetition as the training for a large number of iterations. ClusterClip Sampling outperforms Uniform Sampling by a large margin at the end of the training, which indicates the effectiveness of the clip operation as the training goes on. Surprisingly, both G2S and S2G outperform Uniform sampling at the end of training. S2G sampling even outperforms other sampling methods and shows a tendency to continue increasing the score for longer training. We assume that it benefits from the nature of Open-Orca, as the S2G sampling first learns large clusters followed by other clusters. These large clusters provide dense supervision in specific domains that helps the model quickly align to the instruction-following style in these domains and then transfer to more diverse domains and distributions.

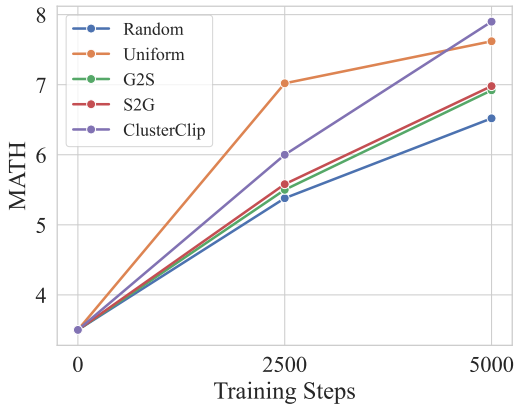


Figure 5: MATH Accuracy of different sampling strategies as the training progresses on the Proof-Pile-2 dataset.

Analysis on Pre-Training However, the results of continual pre-training are different from the results of supervised fine-tuning. As shown in Figure 5, the random sampling under-performs compared with these cluster-based sampling strategies. Uniform sampling outperforms both G2S and S2G sampling methods, which is inconsistent with the results of supervised fine-tuning. It is the data distribution and the repetition rate of data samples in different datasets that lead to the divergence.

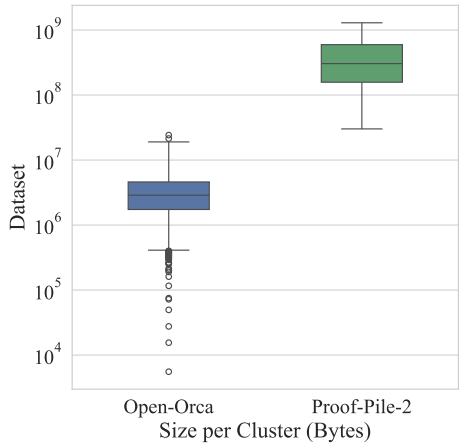
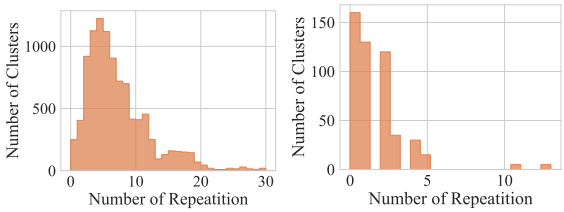


Figure 6: Distribution of cluster size (the number of bytes in each cluster) in different training sets. Clusters of Open-Orca have many outliers, especially tiny clusters, which could affect the sampling performance.

Connection to Data Distribution As the distribution of cluster sizes of Open-Orca and Proof-Pile-2 shown in Figure 6, both training sets have a bell shape that indicates a nearly normal distribution of document sizes, with a long tail of large-size clusters. While Open-Orca has a lot of clusters that have similar sizes, it contains some outlier clusters, like several huge clusters and many tiny clusters. We further calculate the repeated times of samples in these datasets when using Uniform Sampling and visualize the distribution of data repetition in Figure 7. The repetition leads to the model over-fitting on these clusters and side effects on the overall performance of the model. The clusters in Open-Orca have been trained up to more than 30 epochs while the clusters in Proof-Pile-2 have been trained for at most 14 epochs. Thus the overfitting of Uniform Sampling is not severe on Proof-Pile-2, which makes the model generalize well on downstream mathematical tasks. However, the proposed ClusterClip Sampling still outperforms Uniform Sampling in Proof-Pile-2, indicating that the overfitting of rare documents still affects the model per-

formance. The clip operation of ClusterClip effectively mitigates the overfitting of these texts, leading to performance gains on MATH and MMLU tasks.

It is worth-noting that the training order of the samples may need to be carefully scheduled depending on different datasets, based on the results of G2S and S2G Sampling. It can be related to domain-specific learning or curriculum learning (Bengio et al., 2009; Hacohen and Weinshall, 2019; Rozière et al., 2023), which can be investigated in future work. And the cluster-based sampling methods are worth further exploration in this area.



(a) Open-Orca (b) Proof-Pile-2

Figure 7: Distribution of sample repetition on different clusters in Open-Orca and Proof-Pile-2 datasets.

5.3 Ablation of ClusterClip Sampling

We conduct experiments on Proof-Pile-2 to demonstrate how different configuration of ClusterClip Samping affects the overall performance.

Clip Threshold The number of maximum repetitions of one sample in ClusterClip Sampling is primarily manually set to 5, as suggested by Muennighoff et al. (2023). Nonetheless, we aim to verify that our setup is effective in reducing the overfitting on certain clusters. We train Llama2-7B on Proof-Pile-2 datasets with various clip thresholds of repeated samples and keep other setups the same. We provide the results in Figure 8, where the too-small (only one time) or too-large (more than 10 times to repeat) clip thresholds degenerate the performance. Moreover, as the clip threshold increases, the MMLU performance of the trained LLM slightly decreases, which indicates that the overfitting issue becomes worse when samples are repeated more times. Thus, the value of 5 is near optimal for the threshold of the clipping.

The Number of Clusters We aim to investigate the impact of the number of clusters in the application of ClusterClip Sampling methods. We conduct training of Llama2-7B on Proof-Pile-2 datasets with different numbers of clusters, following other

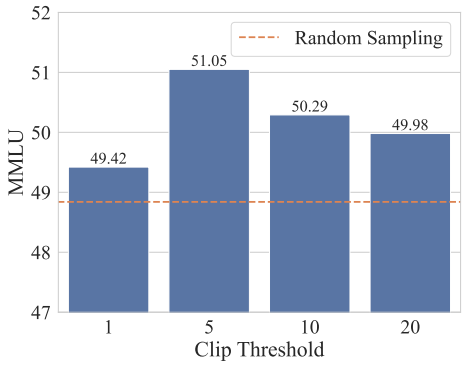


Figure 8: MMLU Accuracy of ClusterClip Sampling when changing the number of clipping thresholds on Proof-Pile-2.

setups in the main experiments. The results are shown in Figure 9, in which the ClusterClip sampling consistently outperforms Uniform sampling by a large margin on the MMLU benchmarks, even though the number of clusters is changing from 100 to 1000, respectively. Moreover, the number of clusters does not affect the performance much under the same cluster-based sampling methods. Thus, the ClusterClip sampling is not sensitive to the number of clusters.

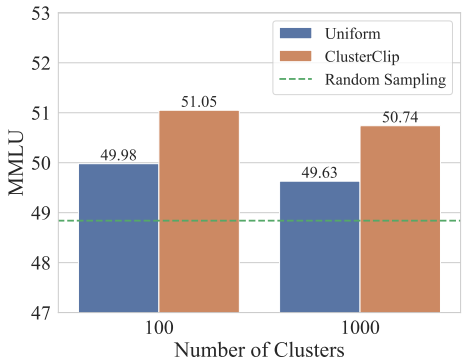


Figure 9: MMLU Accuracy of different sampling methods when changing the number of clusters on Proof-Pile-2 dataset.

6 Conclusion

In this work, we propose **ClusterClip Sampling** based on data clustering to balance the long-tail distribution of the training set of large language models. We compare the proposed method with Random Sampling and other cluster-based sampling method variants in both supervised fine-tuning and pre-training. Extensive experimental results across 7 datasets across diverse tasks and domains demonstrate the effectiveness of ClusterClip Sampling, which outperforms baselines under different training sets and models. We hope this work can instigate more research on data sampling approaches for improving language model training.

Limitations

The ClusterClip Sampling proposed in this work can improve the training of LLMs and mitigate overfitting on small clusters. However, there are limitations in our current work and we hope to enhance the framework of data sampling in future research. Firstly, We use transformer-based sentence embedding to generate data representation and exploit K-Means for clustering. Nonetheless, other data representation or clustering methods can be incorporated. For example, using an LLM to process the texts to achieve better clustering accuracy. Secondly, our current method samples data mainly based on cluster size. However, extra model information or dataset statistics can be incorporated for better sampling strategies. Future work should explore more sophisticated methods to determine the document-level or token-level sampling probabilities. Finally, this study concentrates on language models that only process texts. Training multi-modal generative models that can understand and generate images, videos, and audio poses its challenges. And it requires more sophisticated data processing and sampling techniques, which can be explored in the future.

Ethics Statement

We leveraged data clustering to find semantic clusters of training data and utilize the cluster information for data sampling. There is a risk that the cluster-based sampling aggregates the bias or toxic content in the dataset. However, in this work, we utilize open-sourced datasets to train the LLMs, including Open-Orca (Lian et al., 2023) and Proof-Pile-2 (Azerbayev et al., 2023). These datasets are specifically collected and filtered to avoid toxic and safety issues. When applying the cluster-based sampling strategies for new datasets, data cleaning and filtering methods can be used to remove the harmful content before clustering and training.

References

Amro Abbas, Kushal Tirumala, Daniel Simig, Surya Ganguli, and Ari S. Morcos. 2023. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *CoRR*, abs/2303.09540.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2023. Llemma: An open language model for mathematics. *CoRR*, abs/2310.10631.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM.

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

Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya K. Singh, Pierre H. Richemond, James L. McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Together Computer. 2023. Redpajama: an open dataset for training large language models.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P. Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen S. Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Zhifeng Chen, and Claire

Cui. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 5547–5569. PMLR.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. 2023. Doge: Domain reweighting with generalization estimation. *CoRR*, abs/2310.15393.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, Maximilian Werk, Nan Wang, and Han Xiao. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *CoRR*, abs/2310.19923.

Guy Hacohen and Daphna Weinshall. 2019. On the power of curriculum learning in training deep networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2535–2544. PMLR.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

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

Jean Kaddour. 2023. The minipile challenge for data-efficient language models. *CoRR*, abs/2304.08442.

Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Openorca: An open dataset of gpt augmented flan reasoning traces. `https://https://huggingface.co/Open-Orca/OpenOrca`.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2023. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. *CoRR*, abs/2305.13169.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. 2023. When less is more: Investigating data pruning for pretraining llms at scale. *CoRR*, abs/2309.04564.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2381–2391. Association for Computational Linguistics.

Sören Mindermann, Jan Markus Brauner, Muhammed Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. 2022. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 15630–15649. PMLR.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. 2023. Scaling data-constrained language models. *CoRR*, abs/2305.16264.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of GPT-4. *CoRR*, abs/2306.02707.

Erik Nijkamp, Hiroaki Hayashi, Caiming Xiong, Silvio Savarese, and Yingbo Zhou. 2023. Codegen2: Lessons for training llms on programming and natural languages. *CoRR*, abs/2305.02309.

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

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton-Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2023. Code llama: Open foundation models for code. *CoRR*, abs/2308.12950.

Zhiqiang Shen, Tianhua Tao, Liqun Ma, Willie Neiswanger, Zhengzhong Liu, Hongyi Wang, Bowen Tan, Joel Hestness, Natalia Vassilieva, Daria Soboleva, and Eric P. Xing. 2023. Slimpajama-dc: Understanding data combinations for LLM training. *CoRR*, abs/2309.10818.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13003–13051. Association for Computational Linguistics.

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities. https://github.com/InternLM/InternLM.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S. Morcos. 2023. D4: improving LLM pretraining via document de-duplication and diversification. *CoRR*, abs/2308.12284.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Zige Wang, Wanjun Zhong, Yufei Wang, Qi Zhu, Fei Mi, Baojun Wang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Data management for large language models: A survey. *CoRR*, abs/2312.01700.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2023. Sheared llama: Accelerating language model pre-training via structured pruning. *CoRR*, abs/2310.06694.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. 2023a. Doremi: Optimizing data mixtures speeds up language model pretraining. *CoRR*, abs/2305.10429.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. 2023b. Data selection for language models via importance resampling. *CoRR*, abs/2302.03169.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *CoRR*, abs/2306.05685.

G. K. Zipf. 1949. Human behavior and the principle of least effort.