QuaDMix: Quality-Diversity Balanced Data Selection for Efficient LLM Pretraining

Anonymous ACL submission

Abstract

Quality and diversity are two critical metrics for the training data of large language models (LLMs), positively impacting performance. Existing studies often optimize these metrics separately, typically by first applying quality filtering and then adjusting data proportions. However, these approaches overlook the inherent trade-off between quality and diversity, necessitating their joint consideration. Given a fixed training quota, it's essential to evaluate both the quality of each data point and its complementary effect on the overall dataset. In this paper, we introduce a unified data selection framework called QuaDMix, which automatically optimizes the data distribution for LLM pretraining while balancing both quality and diversity. Specifically, we first propose multiple criteria to measure data quality and employ domain classification to distinguish data points, thereby measuring overall diversity. QuaDMix then employs a unified parameterized data sampling function that determines the sampling probability of each data point based on these quality and diversity related labels. To accelerate the search for the optimal parameters involved in the QuaDMix framework, we conduct simulated experiments on smaller models and use LightGBM for parameters searching, inspired by the RegMix method. Our experiments across diverse models and datasets demonstrate that QuaDMix achieves an average performance improvement of 7.2% across multiple benchmarks. These results outperform the independent strategies for quality and diversity, highlighting the necessity and the framework's ability to balance data quality and diversity.

1 Introduction

The efficiency and preference of pretraining large language models are significantly influenced by the characteristics of the training corpus (Brown et al., 2020; Chowdhery et al., 2023; Longpre et al., 2024). There is evidence from existing research suggesting that the model performance can be improved through the curation of high-quality data (Wettig et al., 2024; Xie et al., 2023b; Sachdeva et al., 2024), the application of data deduplication and diversification strategies (Abbas et al., 2023; Tirumala et al., 2023), and the careful balancing of data distribution across various domains and topics (Liu et al., 2024; Xie et al., 2023a). Nevertheless, identifying optimal configuration of combining those factors remains an open challenge, due to complex interplay between data quality and diversity, which has yet to be fully understood.

There remains two major challenges to identify the optimal data selection strategy. Firstly, the definition of quality and diversity is ambiguous. Previous research has proposed various definitions of quality criteria, including factors such as regular expression (Penedo et al., 2023; Wenzek et al., 2020), educational value (Penedo et al., 2024), similarity to instruction tuning data (Li et al., 2024), etc, each emphasizing only a specific aspect of the data. On the other hand, approaches like (Liu et al., 2024; Abbas et al., 2023) optimize the data mixtures for more effective training, indicating that a better diversity is not necessarily uniform distribution. Secondly, there exists interplay between data quality and diversity. The choice of quality criteria affects the distribution of selected data as illustrated in Figure 1, due to inherent biases in different criteria. Meanwhile, changing of data mixtures influences the data quality, as the quality level differs across different domains. Also, since the high quality data is limited, the trade-off between better quality or diversity is inevitable, which is not feasible by optimizing only for data quality or diversity. How to jointly optimize the data distribution together with the selection of quality criteria remains another unsolved issue.

To address these challenges, we propose a unified data selection framework, QuaDMix, which simultaneously manages data quality and diversity.

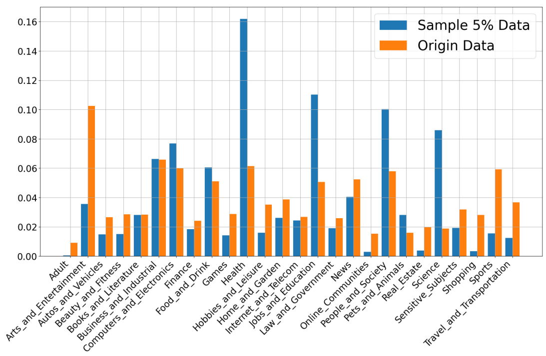

Figure 1: The distribution change of data selected with Fineweb-edu Classifier. With the top5% documents selected, the ratio of certain domains including Health, Jobs and Education, increases for a large margin compared with original data

Firstly, we apply several quality scorers and domain classification on each document in the training corpus, to measure the data quality and diversity. Then a parameterized function is designed to determine the sampling frequency for each document based on those quality and domain labels. Specifically, an aggregated quality score is first computed by weighted averaging the quality scores, where the weights are controlled by adjustable parameters. Then a parameterized sampling function takes the aggregated quality score as input and calculate the sampling frequency, where data with higher quality is assigned with more frequency and the parameters affect how the frequency decreases as the quality diminishes. Here we take the assumption that training samples with higher quality worth sampled for more times. We assign independent parameters for data across different domains to control the diversity via parameters. To find the optimal parameters among the numerous parameter space, we employ a two-step approach inspired by (Liu et al., 2024). First, we train a set of small models on datasets sampled using QuaDMix with various parameter configurations, as an approximation for the performance of larger models. Next, we train a regression model to fit the performance results from this limited set of small models. This regression model is then used to predict the performance for unseen parameter configurations, providing an efficient way to explore the parameter space without exhaustive large-scale training.

To validate the effectiveness of QuaDMix, we train 3000 models with 1M parameters for 1B tokens, each using data sampled from RefinedWeb (Penedo et al., 2023) with various QuaDMix parameters. The optimal parameter configuration is then determined by searching the input space of a trained LightGBM regressor(Ke et al., 2017). We then evaluate different pretraining data selection methods on models with 530M parameters. The optimal configuration identified by QuaDMix achieves superior performance on an aggregated benchmark. Our results also reveal the following insights: (1) Different quality criteria exhibit trade-offs across downstream tasks, but appropriately merging these criteria yields consistent improvements across tasks by leveraging complementary information. (2) The optimal data mixture varies under different quality criteria, indicating the importance of jointly optimizing both the quality and diversity. (3) The target of regression model can guide the preference for specific downstream tasks, enabling task-focused data selection.

2 Related Work

2.1 Pretraining Data Selection

Data quality, diversity, and coverage are critical factors for ensuring the efficiency and generalizability of large language models (Cheng et al., 2024; Touvron et al., 2023; Chowdhery et al., 2023).

To improve data quality, rule-based filtering techniques are commonly employed (Laurençon et al., 2022; Weber et al., 2024; Penedo et al., 2023; Raffel et al., 2020). These methods use handcrafted heuristics, such as removing terminal marks, detecting sentence repetitions, and enforcing length constraints, to exclude low-quality data. While these rules effectively filter out noisy data from the training corpus, they fail to capture semantic-level information, which is crucial for more refined data selection. Alternative approaches aim to address this limitation. For instance, (Wenzek et al., 2020; Marion et al., 2023; Thrush et al., 2024) use model perplexity as a measure of data quality, while (Lin et al., 2025) apply token-level selection by reweighting the loss across tokens. (Xie et al., 2023b) utilize n-gram features to quantify data importance and sample accordingly. Discriminator-based methods (Brown et al., 2020; Du et al., 2022; Gao et al., 2020; Soldaini et al., 2024; Li et al., 2024) select data by comparing it to predefined high-quality datasets, such as Wikipedia or instruction-tuning datasets. However, how much these predefined datasets represent for high-quality relies on empirical judgement. More recently, approaches like (Gunasekar et al., 2023; Sachdeva et al., 2024; Wet-

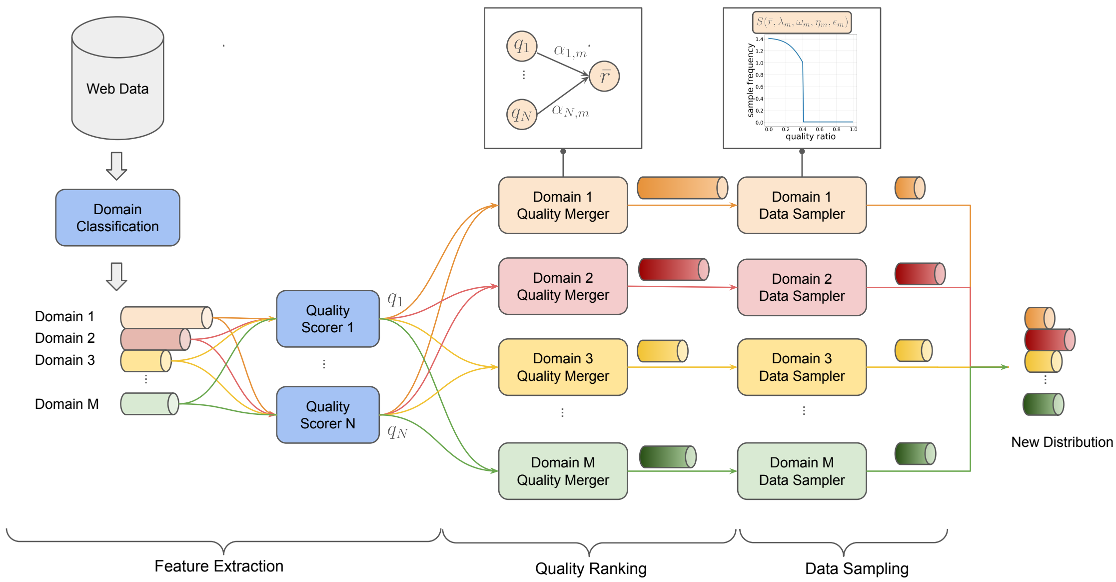

Figure 2: The overall design of QuaDMix. First we extract the data features using classifier and quality scores (QS). Then we calculate quality rank for each domain with the merging parameters. Finally the sampling functions controlled by sampling parameters are applied to generate the final output data.

tig et al., 2024; Penedo et al., 2024) leverage large language models (e.g., GPT-4) to evaluate and filter data based on designed prompts that emphasize various dimensions of value, offering a more nuanced way to define and curate high-quality data.

To optimize data distribution, various methods leverage clustering and representativeness to achieve deduplication and diversification. For example, (Abbas et al., 2023; Shao et al., 2024; Tirumala et al., 2023) employ data clustering techniques to identify and select representative data points, ensuring both diversity and efficiency in the training corpus. Other approaches estimate optimal data mixtures through iterative modeling. (Xie et al., 2023a) first train a small reference model and subsequently optimize the worst-case loss across domains by training a proxy model to identify the optimal data mixture. Similarly, (Bai et al., 2024; Yu et al., 2024; Fan et al., 2024; Gu et al., 2024) calculate influence scores by tracking first-order gradients on an evaluation set, thereby identifying the most valuable data for training. Additionally, (Liu et al., 2024; Ye et al., 2024) simulate the performance of different data mixtures by training a series of proxy models, enabling the prediction of large-model performance with low compute cost.

2.2 Scaling Laws

Neural Scaling Laws have been shown to effectively predict performance across varying training budgets, model sizes, and dataset scales in LLM pretraining (Kaplan et al., 2020; Rae et al., 2022). However, in practical scenarios where dataset size is limited, or data mixtures vary, scaling laws exhibit significant variations (Hoffmann et al., 2022). Several studies have extended scaling laws to account for these complexities. (Muennighoff et al., 2023; Hernandez et al., 2022) explore the impact of data repetition levels on scaling behaviors, while (Ge et al., 2024) investigate scaling dynamics under different domain proportions and dataset sizes. To optimize data compositions, (Liu et al., 2024) propose a regression model for predicting optimal mixtures, and (Kang et al., 2024) further analyze optimal compositions across varying scales. Additionally, (Que et al., 2024) focus on identifying the best data mixtures for the continued pretraining stage, providing insights into refining pretraining strategies under diverse constraints.

3 Methodology

Our approach can be illustrated in 4 parts: 1) We propose the QuaDMix framework, which utilizes a unified parameterized function to govern the data sampling process. 2) We conduct small-scale experiments to explore how different parameter settings within QuaDMix affect the performance of LLM. 3) We train a regression model to capture these effects, using it to identify the optimal parameters. 4) With the optimal parameter settings, we sample large-scale data and train a large language model.

3.1 Design of QuaDMix

We design QuaDMix as a sampling algorithm that simultaneously accounts for data quality and diversity, as shown in Figure 2. Given a pretraining dataset X, we define a sampling function $S(x, \boldsymbol{q}_x, d_x; \boldsymbol{\theta})$, which determines the expected sampling times of each data point x based on its data feature $\boldsymbol{q}_x$ and d_x. Here $\boldsymbol{q_x}$ represents the quality score vector, which includes multiple quality criteria, and d_x denotes the domain to which x belongs. $\boldsymbol{\theta}$ are the parameters to be optimized. The output of this function is fractional value, e.g. $a.b$, meaning the document will be sampled for a times plus another random sampling with probability b.

Feature Extraction To measure a sample's contribution to diversity and its quality, we propose using domain classification and N quality scorers to label the pretraining data. Specifically, we use a domain classifier to divide the dataset into M domains, where x will be assigned a domain label d_x. Then we use N quality scorers to compute the quality vector $\boldsymbol{q_x} = (q_{1,x}, ..., q_{N,x})$, and for each $q_{n,x}$, a smaller value indicates a better quality on that dimension. For the sake of simplicity, we omit x in the subscript in the following discussion.

Quality Ranking We first define a merging function that integrates the scores from various quality filters, aiming to incorporate complementary information provided by different criteria. Assuming there are N criteria, for any individual example $\boldsymbol{x}$ belonging to domain m, the merged quality score is calculated by

$$\bar{q} = \sum_{n=1}^{N} \sigma(q_n)\alpha_{n,m}, \tag{1}$$

where $\boldsymbol{\alpha}_m$ are the merging parameters for domain m. We utilize separate merging parameters to balance the quality criteria across different domains, as the criteria exhibit varying preferences depending on the domain. σ is a normalization function to align the scales of quality criteria.

We then sort the data based on the merged quality score. The sorting is operated separately in each domain. The merged quality rank $\bar{r}$ is calculated by computing the percentile of the data within that domain. That is

$$\bar{r} = \frac{|\{x|d_x = m, \bar{q}_x <= \bar{q}\}|}{|\{x|d_x = m\}|}. \tag{2}$$

Here we calculate the size of the set by adding up the number of tokens for all sample within the set. For a given example in domain m with $\bar{r} = 0.05$, this means that 95% of the tokens in that domain have a worse quality compared to this example. (Note that we use smaller quality scores to represent higher quality.)

Quality Sampling Next, we define the sampling function. We take the assumption that higher-quality data should be sampled more frequently in the final dataset. This assumption is supported by evidence (Penedo et al., 2024), which demonstrates that applying a higher quality threshold improves downstream performance. For any example in domain m with merged quality rank $\bar{r}$, the value of the sampling function is determined by

$$S(\bar{r}) = \begin{cases} (\frac{2}{1+e^{-\lambda_m(\omega_m - \bar{r})}})^{\eta_m} + \epsilon_m, & \bar{r} <= \omega_m \\ \epsilon_m, & \bar{r} > \omega_m \end{cases} \tag{3}$$

We denote $\boldsymbol{\beta}_m = (\lambda_m, \omega_m, \eta_m, \epsilon_m)$ as the sampling parameters for domain m. We use a format of sigmoid to ensure the sampling value is monotonically decreasing as the quality rank goes up (worse quality) and λ_m is used to adjust how fast it decreases. ω_m controls the quality percentile threshold, determining the minimum quality level we aim to retain. η_m is a scaling parameter that adjusts the sampling values, while ϵ_m introduces randomness to incorporate data from all quality ranges. By applying different sampling parameters across domains, we achieve flexible control over domain proportions.

In summary, by integrating (1),(2), and (3), we define the sampling function for individual domain m, with the parameters structured as $\boldsymbol{\theta}_m = (\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m)$. The total number of parameters is $(N + 4) \times M$, where N represents the number of used quality criteria and M denotes the total number of distinct domains.

3.2 Proxy Model Experiments

We first sample a set of values for each parameter defined above, subsequently generating corresponding datasets using the QuaDMix sampling function. Following this, a series of small proxy models are trained on each dataset and evaluated on the validation set to compute the validation loss.

Parameter Sampling The parameter space requires careful design to encompass valuable regions, while avoiding extreme conditions. We sample from the parameter space as following:

Algorithm 1 Parameter Sampling for QuaDMix

Ensure: $\boldsymbol{\theta}$
Require: N, M
  Sample $(a_1, ..., a_N) \sim U(0,1)$
  $\tilde{a}_n = \frac{a_n}{\sum_i a_i}$
  for $m = 1$ **to** M **do**
    Sample $(b_{1,m}, ..., b_{N,m}) \sim U(0,1)$
    $\tilde{b}_{n,m} = \frac{\tilde{a}_n b_{n,m}}{\sum_i \tilde{a}_i b_{i,m}}$
    $\boldsymbol{\alpha}_m = (\tilde{b}_{n,m}), n = 1, ..., N$
    Sample $(\lambda_m, \omega_m, \eta_m, \epsilon_m) \sim U(0,1)$
    $\tilde{\lambda}_m = 10^{3\lambda_m}, \ \tilde{\omega}_m = 0.1\omega_m$
    $\tilde{\eta}_m = \eta_m, \ \tilde{\epsilon}_m = \epsilon_m/1000$
    $\boldsymbol{\beta}_m = (\tilde{\lambda}_m, \tilde{\omega}_m, \tilde{\eta}_m, \tilde{\epsilon}_m)$
    $\boldsymbol{\theta}_m = (\boldsymbol{\alpha}_m, \boldsymbol{\beta}_m)$
  end for
  $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M)$

In the algorithm above, we introduce a global weight for each quality criteria, with the final weight computed by multiplying the global weight by the domain-specific weight. Without this global weight, the expected average weight across domains for each quality criterion would always be $1/N$, which fails to account for the scenario where one quality criterion may suppress another overall. For $\boldsymbol{\beta}_m$, we rescale them accordingly to ensure domain proportions and quality thresholds remain within a reasonable range. Using this process, we generate 3,000 sets of parameters $\boldsymbol{\theta}_i$ and then sample with QuaDMix from our training data, producing 3,000 proxy datasets, denoting as D_i.

Proxy Model Training Next we train the proxy models on each proxy datasets from scratch.

$$f_i^* = \arg\min_f L(f, D_i)$$

After training, we evaluate the proxy models by calculating the loss on the target evaluation datasets.

$$L_i = L(f_i^*, D_{eval})$$

3.3 Parameter Optimizing

Regression Model Fitting The next step is to determine the correlation between the sampled QuaDMix parameters and model performance. We formulate this as a regression problem, as proposed in (Liu et al., 2024), with the goal of learning a function that predicts the target value based on the input features. Specifically, we optimize a regressor R with

$$R^* = \arg\min_R \sum_i ||R(\boldsymbol{\theta}_i) - L_i||^2$$

We evaluate different types of regressors and select LightGBM (Ke et al., 2017), which ensembles multiple decision trees, to predict the target value.

Optimal Parameter Estimation Once the regressor is trained, we search the input space to find the optimal parameters that minimize the predicted loss. Rather than performing a random search across the entire space, we sample 100,000 data points using the algorithm outlined in Section 3.2 to mitigate the influence of outliers on the regressor. To further reduce the variance in the regression predictions, we sort the data points based on their predicted target values and calculate the average of the top 10 data points to determine the final output.

3.4 Large-scale Model Experiments

We then use the optimal parameters to generate large-scale datasets for training large-scale models. In practice, since sorting the quality scores across the entire dataset is computationally expensive, we estimate the quality percentile by randomly selecting a subset of 10,000 documents. Within this subset, we calculate the mapping between the quality percentile and quality score, and then apply this mapping to the entire dataset.

4 Experiments on Regression Model

4.1 Experiment Setup

Datasets We conduct our experiment on RefinedWeb (Penedo et al., 2023). It is an English large-scale dataset for the pretraining of large language models and consists of over 570B(billion) tokens. For the small proxy datasets, we sample it from a subset of RefinedWeb, each containing 1B tokens.

Feature Extraction We generate the necessary data features including data quality and domain index with 3 individual quality filters, AskLLM (Sachdeva et al., 2024), Fineweb-Edu (Penedo et al., 2024), DCLM (Li et al., 2024) and 1 domain classifier (Jennings et al.), which classify the data into 26 different domains with a Deberta V3 (He et al., 2023) architecture.

Training and evaluation For the proxy models, we train them on the proxy datasets for 1B tokens, taking 1 NVIDIA H100 GPU hour and calculate the loss on the validation datasets. To construct the validation datasets, we sample from the instruction-formatted dataset OpenHermes 2.5

Methods	Selected Token	Reading Comprehension	Commonsense Reasoning	Knowledge	Math	Average
Random Selection	500B	32.9	51.6	17.4	2.8	32.3
DSIR	72B	34.9	49.2	17.5	6.9	32.7
RegMix	500B	35.5	52.4	17.7	3.5	33.6
Fineweb-edu	30B	41.4	55.5	20.1	6.0	37.4
AskLLM	30B	38.9	54.2	19.0	2.3	35.5
DCLM	30B	41.2	53.1	19.8	8.2	36.7
Criteria Mix	74B	40.1	53.7	20.0	3.1	36.0
QuaDMix-OH	30B	44.0	55.7	21.0	10.2	39.0
QuaDMix-BMK	30B	**44.8**	**55.7**	**21.3**	**11.5**	**39.5**

Table 1: QuaDMix outperforms the methods focusing only on data quality or data mixture. With benchmark training set as the target, the results further boost.

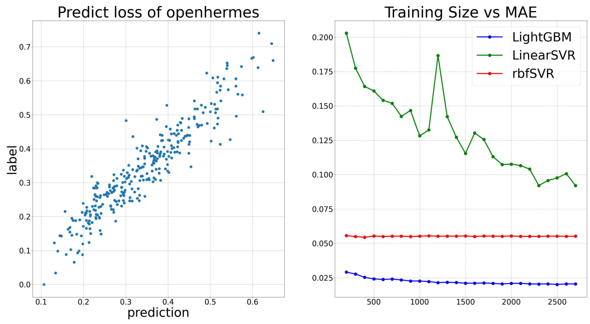

Figure 3: Left: The prediction model loss vs real model loss. Right: The regression model performance (MAE) vs training size.

(Teknium, 2023). As demonstrated in (Li et al., 2024), this dataset is used to train a robust quality filter. To improve efficiency, we sampled 10k samples from it to form a validation subset, named openhermes-10k. Additionally, we test on the training data from the downstream tasks including HellaSwag, ARC-E, ARC-C, MMLU, and TriviaQA to demonstrate the model's ability to optimize for specific downstream tasks by altering the target evaluation datasets.

For the regression model, we split the data into 2800/200 for training and validation. We use Mean Absolute Error (MAE) as the evaluation metric, which calculates the average absolute differences between predicted and actual values.

4.2 Results

We show the results of regression models in Figure 3. The left figure shows strong correlation between the predicted loss and the real model loss (evaluated on OpenHermes) on the validation set, providing the evidence that there exists statistical pattern between the QuaDMix parameters and the model performance. We compare three regression models in the right figure. We can see LightGBM yields better accuracy in predicting the model performance than SVR (Drucker et al., 1996) with Linear kernel and RBF kernel. Also, with larger training size, the accuracy keeps increasing. Considering the training budget, we conduct 3000 proxy experiments in total to get a better results.

5 Experiments on Language Model

In this section we compare different methods of data selection and mixture with QuaDMix by training language models from scratch and evaluating on various downstream tasks.

5.1 Experiment Setup

Training and evaluation We train the language model with 530M parameters from scratch for 500B tokens, taking 32 NVIDIA GPU for 3 days. We use transformer architecture (Vaswani et al., 2017), SwiGLU (Shazeer, 2020) as the activation function and RoPE embeddings (Su et al., 2024). Then we evaluate the model performance using lm-eval-harness (Gao et al., 2023). We choose 9 downstream tasks, including 3 commonsense reasoning tasks (PIQA (Bisk et al., 2019), HellaSwag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018)), 3 reading comprehension tasks (ARC-E/C (Clark et al., 2018), Triviaqa (Joshi et al., 2017)), 1 math problem solving task (SVAMP (Patel et al., 2021)) and 2 knowledge intensive tasks (MMLU (Hendrycks et al., 2021), NQ-open (Kwiatkowski et al., 2019; Lee et al., 2019)). For each benchmark, we used normalized accuracy as the evaluation metric. Some modifications on the testing logic are applied for numerical stability.

A	F	D	Selected Token	Reading Comprehension	Commonsense Reasoning	Knowledge	Math	Average
✓			30B	38.9	53.5	18.6	2.9	35.2
	✓		30B	41.4	55.5	20.1	6.0	37.4
		✓	30B	41.3	53.4	19.7	**12.2**	37.3
✓	✓		30B	41.9	55.6	20.0	5.1	37.5
✓		✓	30B	41.8	54.6	19.8	9.1	37.5
	✓	✓	30B	43.5	55.6	20.8	9.6	38.7
✓	✓	✓	90B	40.7	55.2	19.5	4.6	36.8
✓	✓	✓	180B	37.8	53.9	18.9	2.8	35.1
✓	✓	✓	30B	**44.0**	**55.7**	**21.0**	10.2	**39.0**

Table 2: QuaDMix-OH with different settings on number of quality filters and number of selected tokens. Using all three quality filters yields the best results, and the best practical threshold for quality score is top 5% tokens

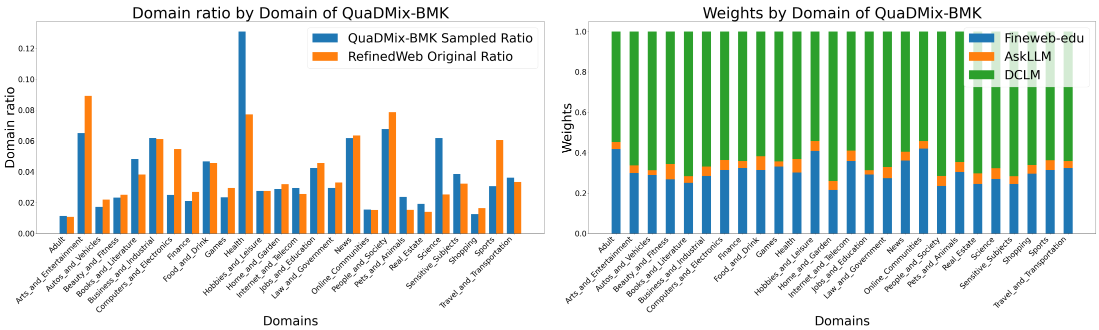

Figure 4: The visualization of optimal parameters from QuaDMix-BMK

5.2 Data Selection Methods

We generate the training data from the RefinedWeb dataset using following data selection methods.

- **Random Selection**: Documents are randomly selected from the whole dataset.
- **Fineweb-edu Classifier**: Documents are scored with Fineweb-edu Classifier (Penedo et al., 2024) with top-k selection
- **AskLLM**: Documents are scored with the probability of generating "Yes" from a prompted large language model (Sachdeva et al., 2024). The top-k documents are selected.
- **DCLM**: Documents are scored with fasttext based classifier (Li et al., 2024) with top-k selection.
- **Criteria Mix**: Following (Wettig et al., 2024), the selected data from the above three filters are merged, with duplicated documents removed.
- **DSIR**: Documents are sampled based on the importance calculated with the N-gram features (Xie et al., 2023b).
- **RegMix**: Following (Liu et al., 2024), we conduct 512 1M porxy experiments and randomly select data using the optimized data mixtures.
- **QuaDMix-OH**: Documents are sampled with the proposed QuaDMix, where Openhermes is used as the validation set for the proxy experiments
- **QuaDMix-BMK**: Documents are sampled with the proposed QuaDMix, where the training set of 5 downstream tasks (HellaSwag, ARC-E, ARC-C, MMLU, TriviaQA) are used as the validation set to generate the optimal QuaDMix parameters.

5.3 Results

The results are summerized in Table 1. We can see that QuaDMix outperforms the methods focusing only on data quality or data mixture on all the benchmarks, proving the necessity of jointly considering quality and diversity. It also shows that the proxy model experiments can well indicate the performance on large scale model. With loss of the benchmark training set as the target when training the regression model, the results further boost. This prove the ability of QuaDMix of optimizing for specific downstream tasks by choosing evaluation datasets in proxy experiments which are more related to the downstream tasks.

Analysis of optimal QuaDMix parameters We show the optimal data mixtures and merging parameters of quality filters from QuaDMix-BMK in

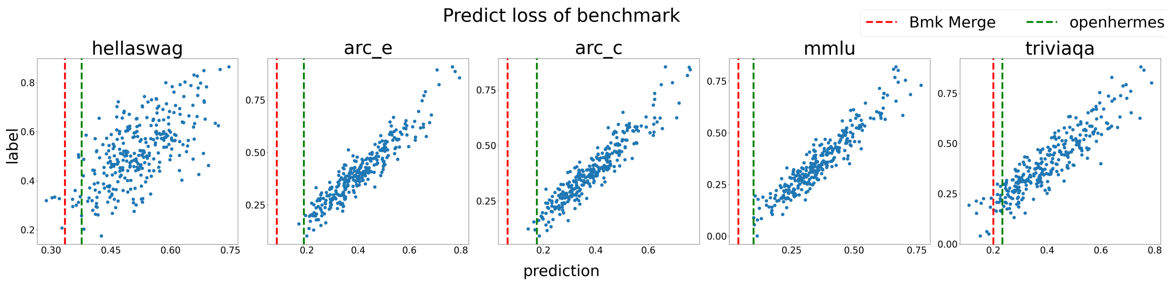

Figure 5: The prediction loss of QuaDMix-BMK surpasses QuaDMix-OH on all 5 downstream tasks.

Method	HellaSwag	ARC-C	ARC-E	MMLU	TriviaQA
QuaDMix-OH	**56.5**	39.2	71.1	34.1	21.6
QuaDMix-BMK	56.1	**40.2**	**71.3**	**34.4**	**22.8**

Table 3: QuaDMix-OH vs QuaDMix-BMK on 5 downstream tasks. The trend mostly agree with the prediction loss on proxy model except for HellaSwag.

Figure 4. We see that the Health and Science domain are upsampled for large margin, while Sports and Computers downsampled, indicating that the downstream tasks we choose have preference for specific domains. The right figure shows that the DCLM quality filter contributes most to the merged quality score, while AskLLM only occupies a small weight among the three filters.

6 Ablations

Quality Merging Benefits Selection To prove the necessity of quality score merging, we select different combinations of quality filters by manually setting the weight of certain filters to 0 when finding the optimal QuaDMix parameters. As shown in Table 2, merging with all three quality filters shows the best performance. Although using one quality filter can be optimal for one specific task, for example DCLM-only for MATH, the merging process reduces intrinsic bias within the quality filters and outperforms in general ability, which is essential for language model pretraining.

More Tokens not always good We also experiment with selecting more tokens by loosing the sampling parameter ω in QuaDMix. In that way we introduce more diversed tokens but lower quality into the training. The results show that selecting 30B tokens, i.e. documents with top5% quality yields the best result, meaning that curing data quality contributes more than increasing the number of unique tokens within this range.

Proxy Ability of Small Models How well the prediction loss on proxy models forecasts the performance on large-scale models is the key factor of QuaDMix. To study this, we train 5 separate regression models, each using the loss on training set of one benchmark as the target. The results on the validation set are shown as blue points in Figure 5. We notice that HellaSwag has larger variance than others, which indicates there may be more influencing factors related with HellaSwag, making the loss on it harder to predict. Then we predict the loss for optimal parameters from QuaDMix-OH and QuaDMix-BMK using each regression model as shown in Figure 5. It is reasonable to see the loss of QuaDMix-BMK surpasses QuaDMix-OH on all tasks since QuaDMix-BMK utilizes benchmark training set as optimizing target. Finally we report the performance of large model in Table 3. Except for HellaSwag, QuaDMix-BMK outperforms QuaDMix-OH on other tasks, which agrees with the trend on prediction loss. The inconsistent conclusion on HellaSwag is because the predict loss has larger variance as mentioned above, making the proxy ability lower than other tasks. How to further increase the proxy ability is one of the future direction to explore.

7 Conclusion

In this paper, we propose a novel data selection method QuaDMix that jointly optimizes the data quality and diversity for language model pretraining. We design a parameterized space that controls both the data quality and diversity, and conduct proxy experiments to find the correlation between the parameter and model performance. The training data generated with optimal parameters are proved to outperform others on various downstream tasks.

8 Limitations

We note several limitations of our work. There exist improvement space for the design of parameter space of QuaDMix. For example the parameters of sampling function may generate similar functions under different parameters, which will cause redundancy and introduce uncertainty into the regression model. Secondly, the searching in the parameter space for optimal parameters is inefficient. We use random guessing in a space with 200 more dimensions, for certain the current optimal parameter is a local minimum and how to effectively search in the parameter space remains unclear. Finally, the proxy ability of small models is crucial, what is the systematic way to improve it is an important yet less explored topic. However, QuaDMix provides a useful solution for jointly optimize for data quality and diversity, and it worth continually exploring on the limitations mentioned above.

References

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S. Morcos. 2023. Semdedup: Data-efficient learning at web-scale through semantic deduplication. Preprint, arXiv:2303.09540.

Tianyi Bai, Ling Yang, Zhen Hao Wong, Jiahui Peng, Xinlin Zhuang, Chi Zhang, Lijun Wu, Jiantao Qiu, Wentao Zhang, Binhang Yuan, and Conghui He. 2024. Multi-agent collaborative data selection for efficient llm pretraining. Preprint, arXiv:2410.08102.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. In AAAI Conference on Artificial Intelligence.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, and et al. 2020. Language models are few-shot learners. In Advances in Neural Information Processing Systems, volume 33, pages 1877–1901.

Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. 2024. Instruction pre-training: Language models are supervised multitask learners. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 2529–2550, Miami, Florida, USA. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, and et al. 2023. Palm: Scaling language modeling with pathways. Journal of Machine Learning Research, 24(240):1–113.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. Preprint, arXiv:1803.05457.

Harris Drucker, Christopher J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. 1996. Support vector regression machines. In Advances in Neural Information Processing Systems, volume 9. MIT Press.

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, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2022. GLaM: Efficient scaling of language models with mixture-of-experts. In Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pages 5547–5569. PMLR.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. 2024. Doge: Domain reweighting with generalization estimation. Preprint, arXiv: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. 2020. The pile: An 800gb dataset of diverse text for language modeling. Preprint, arXiv:2101.00027.

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. 2023. A framework for few-shot language model evaluation.

Ce Ge, Zhijian Ma, Daoyuan Chen, Yaliang Li, and Bolin Ding. 2024. Bimix: Bivariate data mixing law for language model pretraining. Preprint, arXiv:2405.14908.

Yuxian Gu, Li Dong, Hongning Wang, Yaru Hao, Qingxiu Dong, Furu Wei, and Minlie Huang. 2024. Data selection via optimal control for language models. Preprint, arXiv:2410.07064.

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. 2023. Textbooks are all you need. Preprint, arXiv:2306.11644.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. Preprint, arXiv:2111.09543.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In ICLR. OpenReview.net.

Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, Scott Johnston, Ben Mann, Chris Olah, Catherine Olsson, Dario Amodei, Nicholas Joseph, Jared Kaplan, and Sam McCandlish. 2022. Scaling laws and interpretability of learning from repeated data. Preprint, arXiv:2205.10487.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. 2022. An empirical analysis of compute-optimal large language model training. In Advances in Neural Information Processing Systems, volume 35, pages 30016–30030. Curran Associates, Inc.

Joseph Jennings, Mostofa Patwary, Sandeep Subramanian, Shrimai Prabhumoye, Ayush Dattagupta, Vibhu Jawa, Jiwei Liu, Ryan Wolf, Sarah Yurick, and Varun Singh. NeMo-Curator: a toolkit for data curation.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Feiyang Kang, Yifan Sun, Bingbing Wen, Si Chen, Dawn Song, Rafid Mahmood, and Ruoxi Jia. 2024. Autoscale: Automatic prediction of compute-optimal data composition for training llms. Preprint, arXiv:2407.20177.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. Preprint, arXiv:2001.08361.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. Transactions of the Association for Computational Linguistics, 7:452–466.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, Jörg Frohberg, Mario Šaško, , and et al. 2022. The bigscience roots corpus: A 1.6tb composite multilingual dataset. In Advances in Neural Information Processing Systems, volume 35, pages 31809–31826. Curran Associates, Inc.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, and et al. 2024. Datacomp-lm: In search of the next generation of training sets for language models. Preprint, arXiv:2406.11794.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. 2025. Rho-1: Not all tokens are what you need. Preprint, arXiv:2404.07965.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. 2024. Regmix: Data mixture as regression for language model pre-training. Preprint, arXiv:2407.01492.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2024. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 3245–3276, Mexico City, Mexico. Association for Computational Linguistics.

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. Preprint, arXiv: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, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. 2023. Scaling data-constrained language models. In Advances in Neural Information Processing Systems, volume 36, pages 50358–50376. Curran Associates, Inc.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2080–2094, Online. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. Preprint, arXiv:2406.17557.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data only. In Advances in Neural Information Processing Systems, volume 36, pages 79155–79172. Curran Associates, Inc.

Haoran Que, Jiaheng Liu, Ge Zhang, Chenchen Zhang, Xingwei Qu, Yinghao Ma, Feiyu Duan, Zhiqi Bai, Jiakai Wang, Yuanxing Zhang, Xu Tan, Jie Fu, Wenbo Su, Jiamang Wang, Lin Qu, and Bo Zheng. 2024. D-cpt law: Domain-specific continual pre-training scaling law for large language models. Preprint, arXiv:2406.01375.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, and et al. 2022. Scaling language models: Methods, analysis insights from training gopher. Preprint, arXiv:2112.11446.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140):1–67.

Noveen Sachdeva, Benjamin Coleman, Wang-Cheng Kang, Jianmo Ni, Lichan Hong, Ed H. Chi, James Caverlee, Julian McAuley, and Derek Zhiyuan Cheng. 2024. How to train data-efficient llms. Preprint, arXiv:2402.09668.

Yunfan Shao, Linyang Li, Zhaoye Fei, Hang Yan, Dahua Lin, and Xipeng Qiu. 2024. Balanced data sampling for language model training with clustering. Preprint, arXiv:2402.14526.

Noam Shazeer. 2020. Glu variants improve transformer. Preprint, arXiv:2002.05202.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, and et al. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. Neurocomputing, 568:127063.

Teknium. 2023. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants. In huggingface.

Tristan Thrush, Christopher Potts, and Tatsunori Hashimoto. 2024. Improving pretraining data using perplexity correlations. Preprint, arXiv:2409.05816.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S. Morcos. 2023. D4: Improving llm pretraining via document de-duplication and diversification. Preprint, arXiv:2308.12284.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc.

Maurice Weber, Daniel Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. Preprint, arXiv:2411.12372.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 4003–4012, Marseille, France. European Language Resources Association.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. Qurating: Selecting high-quality data for training language models. Preprint, arXiv:2402.09739.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2023a. Doremi: Optimizing data mixtures speeds up language model pretraining. In Advances in Neural Information Processing Systems, volume 36, pages 69798–69818. Curran Associates, Inc.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. 2023b. Data selection for language models via importance resampling. In Advances in Neural Information Processing Systems, volume 36, pages 34201–34227. Curran Associates, Inc.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Yunhua Zhou, Jun Zhan, and Xipeng Qiu. 2024. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. Preprint, arXiv:2403.16952.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.