ScalingFilter: Assessing Data Quality through Inverse Utilization of Scaling Laws

Ruihang Li[1,2,5,6], **Yixuan Wei**[2,3], **Miaosen Zhang**[2,4],
Nenghai Yu[1,5,6]*, **Han Hu**[2], **Houwen Peng**[2]*
[1]School of Cyber Science and Technology, University of Science and Technology of China
[2]Microsoft Research Asia [3]Tsinghua University [4]Southeast University
[5]Anhui Province Key Laboratory of Digital Security
[6]the CAS Key Laboratory of Electromagnetic Space Information
{ruihangli@mail., ynh@}ustc.edu.cn wei-yx20@mails.tsinghua.edu.cn miazhang@seu.edu.cn
ancientmooner@gmail.com penghouwen@icloud.com

Abstract

High-quality data is crucial for the pre-training performance of large language models. Unfortunately, existing quality filtering methods rely on a known high-quality dataset as reference, which can introduce potential bias and compromise diversity. In this paper, we propose ScalingFilter, a novel approach that evaluates text quality based on the perplexity difference between two language models trained on the same data, thereby eliminating the influence of the reference dataset in the filtering process. An theoretical analysis shows that ScalingFilter is equivalent to an inverse utilization of scaling laws. Through training models with 1.3B parameters on the same data source processed by various quality filters, we find ScalingFilter can improve zero-shot performance of pre-trained models in downstream tasks. To assess the bias introduced by quality filtering, we introduce *semantic diversity*, a metric of utilizing text embedding models for semantic representations. Extensive experiments reveal that *semantic diversity* is a reliable indicator of dataset diversity, and ScalingFilter achieves an optimal balance between downstream performance and semantic diversity. [1]

1 Introduction

The success of large language models (LLMs) is significantly influenced by the quality and quantity of the pre-training corpus. Researchers have developed various data curation pipelines to enhance dataset quality, focusing on raw web crawling, text extraction, repetition and toxic content removal, and, notably, quality filtering (Brown et al., 2020; Rae et al., 2021; Penedo et al., 2023).

Quality filters aim to extract high-quality data from a noisy raw corpus, thereby improving the language model's performance without increasing training costs. Existing filters are broadly classified into two categories: reference-dependent and reference-free approaches. Reference-dependent methods, such as binary classification (Brown et al., 2020; Gao et al., 2020; Chowdhery et al., 2023) and DSIR (Xie et al., 2023), filter out low-quality data by comparing it with high-quality seed datasets. While effective, these methods inevitably introduce biases present in the reference data, such as specific writing styles or topics, thereby limiting the diversity and representativeness of training corpus (Soldaini et al., 2023). In contrast, reference-free methods, such as perplexity gating (Marion et al., 2023), assess data quality using predefined metrics like perplexity scores from pre-trained models. These methods mitigate the biases introduced by reference datasets but encounter challenges due to the indirect relationship between absolute perplexity and document quality. This indirect relationship inadvertently favors data with simple and repetitive content. Although such content is easier for models to predict, it contributes minimally to learning diversity and complexity (Wettig et al., 2024).

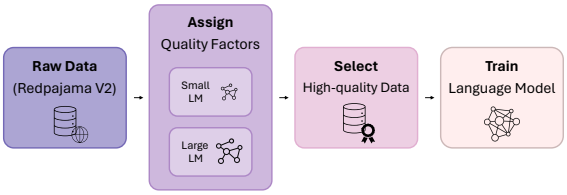

Figure 1: In ScalingFilter, we assess the quality of text documents by their scaling characteristics with language models in different sizes.

To address these issues, we introduce a simple yet effective quality filtering approach named ScalingFilter, which inversely leverages recent scaling laws in generative modeling to assess data quality. The core idea is to analyze the perplexity differences between two pre-trained models on the same data and assess the data quality based on these dif-

*Corresponding authors.
[1]Our code is available at https://github.com/scalingfilter/scalingfilter

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 3209–3222
November 12-16, 2024

ferences. We find a positive correlation between data quality and perplexity differences by inversely deriving Chinchilla scaling law (Hoffmann et al., 2022). In other words, given a pair of pre-trained models of different sizes, documents with higher perplexity differences indicate higher quality.

ScalingFilter involves utilizing the difference between two separate models for data quality assessment, effectively addressing the bias issue induced by relying on a single model trained on the reference data. This approach also mitigates the problem of selecting simple and repetitive texts that arise from overfitting to the perplexity metric, thereby enhancing data diversity and complexity. Furthermore, ScalingFilter offers a theoretical analysis for using perplexity differences as a quality indicator for data filtering by inversely deriving model scaling laws.

Our experiments demonstrate that ScalingFilter is superior to existing methods in improving filtered data quality while preserving data diversity. Specifically, we employ a pair of meta-models with sizes of 124M and 774M parameters to assess the perplexity difference for each document in the raw dataset, and then select the high-quality ones using a top-k strategy. We train a 1.3B model from scratch using filtered high-quality data. We then evaluate its zero-shot performance on downstream tasks and assess the semantic diversity of the filtered dataset.

The results demonstrate that ScalingFilter outperforms the unfiltered baseline and previous state-of-the-art (SoTA) quality filtering methods. Specifically, compared to the unfiltered baseline, ScalingFilter achieves a +3.09% improvement in downstream accuracy and a +2.23 increase in semantic diversity. When compared with perplexity gating (Marion et al., 2023; Wenzek et al., 2019), ScalingFilter achieves a +1.12% improvement in performance and a +4.7 increase in semantic diversity.

In summary, the contributions of this work are threefold:

1. We introduce *quality factor*, a novel metric that correlates directly with the quality of training data through the lens of model scaling laws, offering a more precise and unbiased approach to data curation.

2. We propose ScalingFilter, a new quality filtering method that utilizes the *quality factor* to curate high-quality datasets without relying on reference data, thereby mitigating the risk of bias and enhancing the representativeness of the training corpus.

3. To evaluate the data diversity of filtered datasets, we introduce *semantic diversity* as a novel and reliable metric. Extensive experiments demonstrate that ScalingFilter more effectively preserves the richness and variety of the raw data compared to conventional quality filtering approaches.

2 Methodology

Overview. Existing quality filtering methods depend on either a reference high-quality dataset or the absolute perplexity scores of documents from a single language model, which can introduce potential bias or result in inferior performance. In this section, we will elaborate on the principles of ScalingFilter through mathematical derivation. The core concept of ScalingFilter lies in estimating the quality of data samples by inversely applying the scaling law. Specifically, the scaling law reveals a power-law decrease in loss with increasing model size or data size (Hestness et al., 2017; Kaplan et al., 2020; Hoffmann et al., 2022; Aghajanyan et al., 2023). Ultimately, the scaling law yields the optimal model/data scaling-up allocation strategy. In other words, under the same computational budget (TFLOPS), it determines the optimal ratio of model size to the number of training tokens to achieve the lowest loss, represented by a model scaling exponent a and a data scaling exponent b.

Extensive experiments comparing multiple datasets with known quality differences revealed that high-quality data increases the model scaling exponent a (Bi et al., 2024). Specifically, the experiments compared the early and final versions of in-house data together with OpenWebText2 (Gao et al., 2020), revealing that the final version of OpenWebText2 results in the highest a, while the early version with the poorest quality leads to the lowest a. Intuitively, a higher value of a accelerates the rate at which the loss decreases as the model parameters increase. This positive relationship will be demonstrated later. Such an observation suggests that high-quality data enhances logical clarity and decreases predictive difficulty after adequate training. Consequently, scaling up the model size becomes more beneficial when the compute budget is increased (Bi et al., 2024; Aghajanyan et al., 2023; Kaplan et al., 2020; Hoffmann et al., 2022).

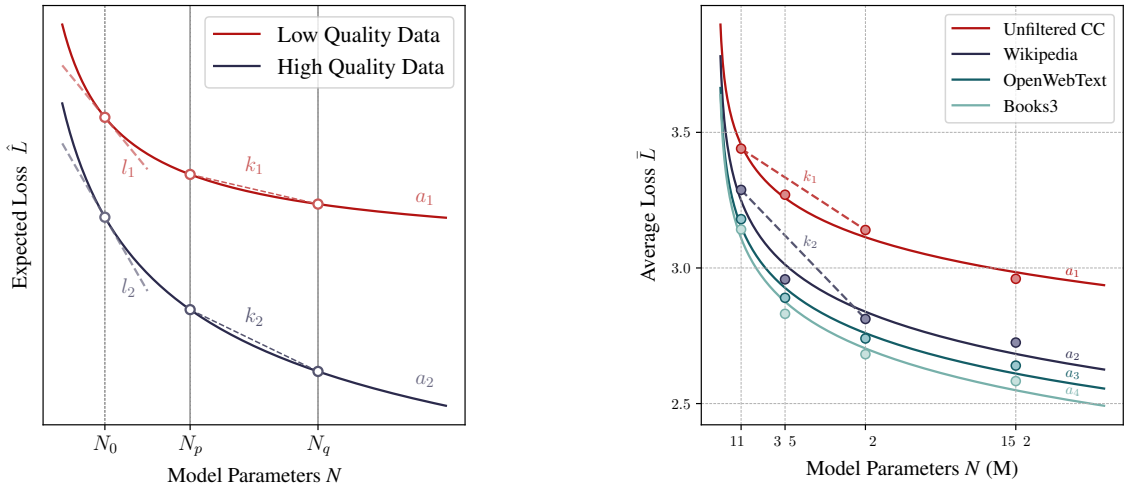

Figure 2: **(a)** A visual diagram illustrates the theoretical result that high-quality data accelerates the rate of loss decrease as model parameters increase, resulting in larger model scaling exponents a. **(b)** We calculated the average loss of GPT-2 models of different sizes on several datasets with recognized quality levels: Wikipedia, OpenWebText, and Books3 represent high-quality data, while Unfiltered CommonCrawl represents low-quality data. The results closely align with the theoretical analysis shown in (a), which indicates that high-quality data accelerates the rate of loss decrease as model parameters increase.

By inversely applying this principle, ScalingFilter estimates data quality based on the rate of loss decrease in models with a fixed parameter difference, thereby separating high-quality data from the raw dataset.

To proceed, we will first define the *quality factor*, which is the magnitude of loss reduction. Then, starting from the formula of the scaling law, we will demonstrate the positive correlation between the *quality factor* and the model scaling exponent a. Furthermore, based on the positive correlation between a and data quality observed in (Bi et al., 2024), we can ultimately prove the positive correlation between the *quality factor* and data quality.

Quality factor. We begin with defining the *quality factor*, which we will later demonstrate to have a positive correlation with data quality. We denote the smaller meta-model as p and the larger one as q. Both meta-models share the same architecture and are trained on the same dataset, with the only difference being the parameter counts: N_p for p and N_q for q, with $N_p < N_q$. Let x_i be a given text sample, and denote the *quality factor* of this sample as d_i. Then, we have:

$$d_i := \frac{\mathrm{PPL}_p(x_i)}{\mathrm{PPL}_q(x_i)} \tag{1}$$

where $\mathrm{PPL}_p(x_i)$ and $\mathrm{PPL}_q(x_i)$ represent the perplexity scores of the text sample x_i when evaluated by p and q, respectively. It's important to note that perplexity has a direct relationship with the cross-entropy loss L because $\mathrm{PPL} = 2^L$, indicating that the perplexity score is positively related to the loss L.

Quality factor is positively correlated with data quality. Next, we will introduce the expression of the scaling law (Hoffmann et al., 2022; Kaplan et al., 2020; Henighan et al., 2020; Aghajanyan et al., 2023) and transform it into a form involving the model scaling exponent a which, as we introduced in the overview, is known to have a positive relationship with data quality (Bi et al., 2024).

Given the number of model parameters N and training tokens D, the expected model loss $\hat{L}$ is formulated (Hoffmann et al., 2022) as:

$$\hat{L}(N, D) = E + \frac{A}{N^\alpha} + \frac{B}{D^\beta} \tag{2}$$

where E represents the minimal achievable loss, corresponding to the entropy of natural text. The terms $\frac{A}{N^\alpha}$ and $\frac{B}{D^\beta}$ account for the functional approximation error and the optimization or convergence error, respectively (Aghajanyan et al., 2023). Here, A, B, α, and β are hyperparameters related to the model architecture and the training data. The scaling law, indicating the optimized numbers of N and D under a given compute budget C, follows a power-law form (Kaplan et al., 2020; Hoffmann et al., 2022):

$$N_{opt} \propto C^a \quad D_{opt} \propto C^b \tag{3}$$

where $a = \frac{\beta}{\alpha+\beta}$ and $b = \frac{\alpha}{\alpha+\beta}$ represent the model and data scaling exponents (Bi et al., 2024; Hoffmann et al., 2022) respectively, indicating the proportions of the total computational budget allocated to model scaling and data scaling in the optimal computation allocation.

Consider setting $\eta \doteq \alpha + \beta$, then $\hat{L}$ can be presented as:

$$\hat{L}(N, D) = E + \frac{A}{N^{(1-a)\eta}} + \frac{B}{D^{a\eta}} \tag{4}$$

We focus on the relationship between expected loss $\hat{L}$ and model scaling exponent a as well as model size N, and thus denote $\hat{L}$ as $\hat{L}(a, N)$. It's obvious that $\hat{L}$ decreases as N increases. We further prove in Appendix A.2.1 that at a specific N_0, the slope of the tangent to the $\hat{L} - N$ curve decreases as a increases (i.e., the larger the a, the steeper the tangent, as illustrated in Figure 2a that l_2 is steeper than l_1). Due to this monotonic relationship, we can infer the value of a from the slope of the tangent: for a given N_0, a steeper tangent (greater absolute value of the slope) indicates a larger a, that is:

$$a \propto -\frac{\partial L}{\partial N}\bigg|_{N=N_0} \tag{5}$$

Furthermore, we prove in Appendix A.2.2 that the above conclusion can be extended from the tangent slope at a given N_0 to the slope of the secant line for any given ΔN (i.e. k_i in Figure 2a). Letting $\Delta N = N_q - N_p$, the slope of the secant line is always negative, and a is positively correlated with the negative slope of the secant line. Since the quality factor d is also positively correlated with the negative slope of the secant line, it follows that d is positively correlated with a:

$$\begin{cases} a \propto -\frac{\hat{L}(N_q)-\hat{L}(N_p)}{N_q-N_p}, \\ d = 2^{\hat{L}(N_p)-\hat{L}(N_q)} = 2^{-(\hat{L}(N_q)-\hat{L}(N_p))} \end{cases}$$
$$\Longrightarrow \; d \propto a \tag{6}$$

Based on empirical observations from (Bi et al., 2024), higher values of a are achieved with high-quality data, indicating that ***quality factor*** d **is positively correlated with data quality**.

This conclusion aligns with our practical comparative tests. As shown in Figure 2b, we calculated the average loss of GPT-2 models of different sizes on several datasets with recognized quality levels: Wikipedia, OpenWebText, and Books3 represent high-quality data, while Unfiltered CommonCrawl represents low-quality data, based on a random sample of 10k documents from each dataset. The results align closely with the theoretical estimates shown in Figure 2a, where the high-quality data shows a steeper secant ($k_2 > k_1$) compared to low-quality data. It's worth noting that a single case might deviate from the training data distribution of the meta-models, leading to higher absolute perplexity for various model sizes. Thus, relying solely on single perplexity as a quality criterion can result in misjudgments. However, high-quality data follows a law where perplexity decreases more with an increase in model parameters, indicating a greater perplexity difference (i.e., *quality factor*).

Selecting high-quality data with *quality factor*. We have demonstrated that the *quality factor* can directly characterize data quality above, so it's straightforward to directly use it to select high-quality data from a noisy unfiltered dataset. We call this simple yet effective method as **ScalingFilter**, as illustrated in Figure 1. Consider a unfiltered set of documents $\mathcal{S}$, containing both high and low-quality documents. For each sample $x_i \in \mathcal{S}$, we calculate the quality factor d_i for it. As derived previously, samples with higher d_i are of better quality. The top-k samples are then selected based on the desired cleaning rate to form the resulting dataset.

3 Experiments

In this section, we will demonstrate the effectiveness of ScalingFilter through extensive experiments. Specifically, language models trained on data filtered by ScalingFilter consistently achieved superior performance across various downstream tasks, compared to the unfiltered baseline and other common quality filtering approaches, highlighting the higher quality of the data. Furthermore, by measuring the semantic diversity of the filtered dataset, we found that ScalingFilter effectively preserved the diversity present in the original dataset.

3.1 Data Quality Evaluation

Setup. We begin with five CommonCrawl dumps from 2019 to 2023, processed through the CCNet (Wenzek et al., 2019) pipeline, in accordance with (Computer, 2023). From the preprocessed dataset, 500 GB of text data are randomly selected as our baseline, yielding approximately

Table 1: Zero-shot downstream accuracy of models trained with different quality filters. We cover a variety of tasks and widely used datasets (Penedo et al., 2023; Brown et al., 2020; Chowdhery et al., 2023; Dey et al., 2023; Biderman et al., 2023), including sentence completion, coreference resolution, natural language inference and multiple-choice question answering. For binary classification (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023) and importance resampling (Xie et al., 2023), we leverage the best results from various reference datasets, whereas perplexity gating (Marion et al., 2023) utilizes the larger model's perplexity in our meta-models.

Quality Filter	Hellaswag	LAMBADA	Winogrande	PIQA	ARC	OpenbookQA	BoolQ	Avg
Random	45.40	41.96	51.07	69.80	39.88	32.40	56.76	48.18
Binary Classification	48.13	**48.96**	54.30	69.75	41.66	30.40	61.38	50.65
Importance Resampling	47.52	48.36	54.38	68.50	41.63	**32.60**	60.80	50.54
Perplexity Gating	48.17	**48.96**	53.04	69.75	41.54	29.60	60.00	50.15
ScalingFilter (Ours)	**49.07**	48.42	**55.09**	**70.57**	**42.67**	31.40	**61.68**	**51.27**

125 billion tokens for additional quality filtering. In each experiment, we train a decoder-only model with 1.3B parameters, using the same model architecture as (Peng et al., 2023). Each model is trained on 25B tokens until performance levels off, according to (Hoffmann et al., 2022; Penedo et al., 2023; Marion et al., 2023), which takes approximately 4 days on 4 nodes with 8 NVIDIA Tesla V100 GPUs. We use pre-trained GPT-2 models (Radford et al., 2019) as default meta-models to calculate quality factors for each sample. The smaller and larger models have 124M and 774M parameters, respectively. We later perform ablation studies to discuss impacts by the pre-training data. Following (Penedo et al., 2023), we utilize the `lm-evaluation-harness` library (Gao et al., 2023) to evaluate zero-shot performance across various downstream tasks of each model trained on documents retained through specific quality filtering method. We encompasses a variety of tasks and widely used datasets (Penedo et al., 2023; Brown et al., 2020; Chowdhery et al., 2023; Dey et al., 2023; Biderman et al., 2023), including sentence completion (Hellaswag (Zellers et al., 2019) and LAMBADA (Paperno et al., 2016)), coreference resolution (Winogrande (Sakaguchi et al., 2021)), natural language inference (ARC (Clark et al., 2018)), and multiple-choice question answering (PIQA (Bisk et al., 2020), OpenbookQA (Mihaylov et al., 2018), BoolQ (Clark et al., 2019)).

Baselines. We compare ScalingFilter with random selection, binary classification (Brown et al., 2020; Gao et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023), importance resampling (Xie et al., 2023) and perplexity gating (Marion et al., 2023). All quality filters will keep 70% of the unfiltered documents in align with (Computer, 2023), if not specified otherwise. As for binary classification, we choose Wikipedia, books and OpenWebText as positive samples and unfiltered CommonCrawl documents as negative ones, following (Du et al., 2022; Chowdhery et al., 2023). We set the shape parameter of Pareto distribution $\alpha = 9$, following (Brown et al., 2020; Gao et al., 2020; Chowdhery et al., 2023). As to importance resampling, we follow the settings in (Xie et al., 2023), with OpenWebText (Gokaslan and Cohen, 2019) as the target dataset. As for perplexity gating, we follow (Marion et al., 2023) as well as our cleaning ratio, keeping documents with perplexity ranging from 15th to 85th percentiles, resulting in keeping the middle 70% documents of the unfiltered dataset. Perplexity is computed by the larger of the meta-models, the one with higher capacity and ability.

Results. Table 1 shows the comparison between various data quality filter methods. In summary:

- On average, ScalingFilter shows a 0.62% improvement over the widely-used binary classification quality filtering method and a 0.73% improvement over importance resampling, achieving the state-of-the-art performance.
- ScalingFilter achieves a 1.12% improvement in average accuracy over perplexity gating, a competing reference-free quality filtering approach.

Notably, for binary classification (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023) and importance resampling (Xie et al., 2023), we use the best results from various reference datasets, specifically OpenWebText. The results for perplexity gating (Marion et al., 2023) use the larger model's perplexity in our meta-models for a fair comparison. Ablations concerning the reference datasets of the aforementioned methods will be discussed in subsequent sections.

Table 2: Ablations on effects of meta-models training data within the ScalingFilter framework. The results reveal that meta-models trained on alternative datasets also showcase competitive performance, indicating that there is not an overly strong dependency on meta-models pretrained on WebText, emphasizing the robustness and flexibility of ScalingFilter variants.

Training Data	Hellaswag	LAMBADA	Winogrande	PIQA	ARC	OpenbookQA	BoolQ	Avg
Unfiltered CC	47.34	47.78	54.22	**70.78**	40.64	30.40	60.95	50.30
Wikipedia	48.81	47.64	**56.67**	69.31	41.71	**32.60**	61.07	51.12
OpenWebText	48.15	46.01	54.06	69.91	43.01	31.40	60.89	50.49
WebText†	**49.07**	**48.42**	55.09	70.57	**42.67**	31.40	**61.68**	**51.27**

† For model pairs trained on WebText, we directly use OpenAI GPT-2 models from HuggingFace, which is the meta-models used in the original ScalingFilter framework.

Meta-models trained on various datasets exhibit competitive and comparable performance. We detail ablation studies with meta-models trained on different datasets in Table 2. Besides the meta-models trained on WebText, results of which are shown in Table 1, we trained meta-models on a subset of Wikipedia, OpenWebText, and unfiltered CommonCrawl data with no more than 25B tokens. Each dataset was used to train meta-models for 1 epoch. The results demonstrate that all experiments outperform the baseline of random selection. Training on unfiltered CommonCrawl or OpenWebText yielded results competitive with those from other quality filtering methods. Furthermore, training on Wikipedia achieved results very close to the best, with a marginal gap of 0.15%.

Ablations on different sizes of meta-models. We perform experiments to investigate impacts brought by using pairs of meta-models with different sizes. The results are briefly presented in Table 4. When using a pair of meta-models with relatively small differences in the number of parameters to estimate the quality factors of data, there is a certain degree of performance degradation in the downstream tasks. Reducing the size of the larger models in meta-models from 774M to 335M decreases the average performance on downstream tasks by 0.96%. Conversely, increasing the size of the smaller models in meta-models from 124M to 335M results in a decrease of 1.28% in performance. This suggests that a larger parameter gap may more effectively amplify differences in how models fit textual data, allowing for a more reliable assessment of the quality factor. Detailed exploration of this hypothesis is left as future work.

Ablations on reference datasets. We also examine the impacts of different reference datasets on popular quality filtering methods that rely on a reference. Results are shown in Table 3. Binary classification using OpenWebText as the positive class results in the best performance, similar to importance resampling with the same dataset as a reference. This aligns with the findings in (Bi et al., 2024), which confirm that OpenWebText has superior data quality. Binary classification with a mixed dataset including OpenWebText, Wikipedia, and books yields inferior results, possibly due to the classifier's training recipe, such as the mixing ratio of the three datasets. Surprisingly, importance resampling with Wikipedia results in similar average accuracy to the random baseline, with much better accuracy in ARC and BoolQ but significantly worse performance in sentence completion tasks like Hellaswag and LAMBADA, possibly due to the serious domain shift towards Wikipedia. In conclusion, the choice of reference datasets has a significant impact on the performance of quality filters that rely on references.

3.2 Data Diversity Evaluation

Training large language models requires diverse data. Current quality filters, by favoring text data similar to the reference dataset, may discard documents on informal topics or from minorities, reducing the trained model's knowledge diversity (Wenzek et al., 2019; Soldaini et al., 2023). How can we assess a dataset's data diversity? We introduce a metric to measure a document group's semantic diversity.

Semantic diversity metric. Following (Friedman and Dieng, 2022), we define semantic diversity as the exponential of the Shannon entropy of the semantic similarity matrix's eigenvalues. For a set of text documents $x_1, x_2, ..., x_n$ and a semantic similarity function f, we obtain a similarity matrix $\mathbf{S}$, where each entry $s_{i,j} = f(x_i, x_j)$. Denoting $\lambda_1, \lambda_2, ..., \lambda_n$ as the eigenvalues of $\mathbf{S}/n$, we define semantic diversity as follows.

$$\text{SemanticDiversity} = \exp\left(-\sum_{i=1}^{n} \lambda_i \log \lambda_i\right) \tag{7}$$

Table 3: Ablation studies on the effects of reference data. We varied the reference datasets for binary classification (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023) and importance resampling (Xie et al., 2023). The results indicate that OpenWebText is the optimal reference dataset choice for both reference-dependent quality filtering methods.

Quality Filter	Hellaswag	LAMBADA	Winogrande	PIQA	ARC	OpenbookQA	BoolQ	Avg
Random	45.40	41.96	51.07	69.80	39.88	32.40	56.76	48.18
Binary Classification								
OpenWebText	**48.13**	**48.96**	54.30	**69.75**	41.66	30.40	61.38	**50.65**
Wikipedia	46.80	46.96	53.35	69.15	41.40	32.00	61.31	50.14
Mixed†	47.10	47.68	53.43	68.61	42.19	32.20	57.71	49.85
Importance Resampling								
OpenWebText	47.52	48.36	**54.38**	68.50	41.63	**32.60**	60.80	50.54
Wikipedia	43.08	38.56	51.93	66.65	**42.76**	32.40	**61.90**	48.18

† This experiment uses a mixed dataset as reference dataset following (Du et al., 2022; Chowdhery et al., 2023), with OpenWebText, Wikipedia, and books. The classification scores are directly obtained from quality signals provided by Redpajama V2.

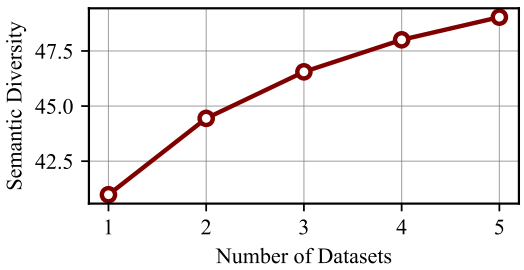

Figure 3: Positive correlation between the number of datasets and semantic diversity, demonstrating semantic diversity as a reliable measure of data diversity.

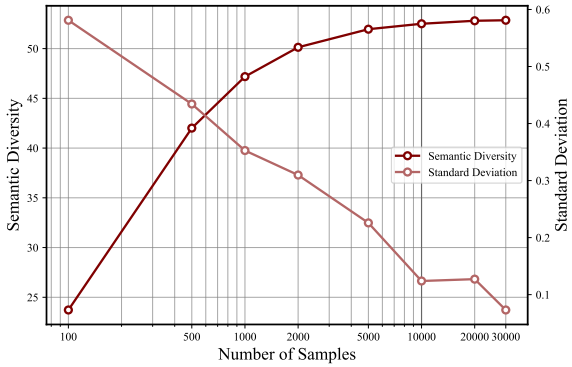

Figure 4: Results on the relationship between semantic diversity and sample size. Semantic diversity stabilizes at a sample size of 10,000, with a standard deviation below 0.2. Therefore, we choose 10,000 as our sample size for calculating semantic diversity, as it represents the dataset's diversity adequately while ensuring computational efficiency.

A pre-trained language model extracts each document's semantic embedding, using cosine similarity as the similarity function. In our experiments, we utilize the `bge-base-en-v1.5` model (Xiao et al., 2023) with the `sentence_transformers` library due to its efficiency and outstanding performance in various text embedding-related retrieval and clustering tasks.

Selecting a proper size of documents. Computational constraints prevent calculating semantic diversity for all documents in the dataset. Experiments on the unfiltered dataset help select an appropriate document count for calculating the semantic diversity metric. For each experiment, we randomly select n samples, calculate their semantic diversity score, and repeat this process 10 times to compute the average score and standard deviation. Results are displayed in Figure 4. Results indicate that semantic diversity stabilizes when the group exceeds 10,000 samples, with a standard deviation of 0.12. We choose 10,000 samples for subsequent experiments to balance accuracy and efficiency.

The proposed metric can reflect data semantic diversity. Our experiments showed that semantic diversity effectively reflects data diversity under multi-datasets settings. We selected five datasets with diverse topics or writing styles, including news articles (CC-News (Hamborg et al., 2017)), movie reviews (IMDB (Maas et al., 2011)), forums (Reddit [2]), Wikipedia, and crawled web pages (OpenWebText (Gokaslan and Cohen, 2019)). We extracted the same number of samples from one or more of the above datasets, creating a mixed subset of 10,000 samples. We then averaged the relationship between semantic diversity and the number of datasets (N). Figure 3 shows a positive correlation between semantic diversity and the number of datasets (N), indicating that semantic diversity accurately reflects data diversity within datasets.

Quality filtering with quality factor keeps the diversity of the unfiltered dataset. We assess the semantic diversity of datasets resulting from various quality filtering methods. The results are presented in Table 5. Most quality filters achieve

[2] https://www.reddit.com

Table 4: Ablations on effects of sizes of meta-models. To note, the original ScalingFilter uses a pair of meta-models with 124M and 774M parameters, respectively.

Small Model	Large Model	HellaSwag	LAMBADA	Winogrande	PIQA	ARC	OpenbookQA	BoolQ	Avg
124M	335M	48.77	47.25	53.67	69.75	41.12	**32.00**	59.60	50.31
335M	774M	48.32	45.76	52.41	70.18	42.05	30.60	60.61	49.99
124M	774M	**49.07**	**48.42**	**55.09**	**70.57**	**42.67**	31.40	**61.68**	**51.27**

higher diversity than the original unfiltered dataset, likely due to the removal of a large number of machine-generated spams with similar semantic meanings. The results indicate that importance resampling achieves the highest diversity, at 56.25, attributed to its resampling strategy. ScalingFilter results in greater diversity compared to the most commonly used binary classification, thanks to its reference-free nature. Perplexity gating reduces the diversity of the original dataset, supporting the conclusion from (Wenzek et al., 2019) that filtering data based solely on perplexity thresholds can introduce unexpected bias to data diversity.

Quality Filter	Diversity
Random	$52.50_{0.12}$
Binary Classification	$53.99_{0.19}$
Importance Resampling	$\mathbf{56.25}_{0.21}$
Perplexity Gating	$50.03_{0.21}$
ScalingFilter	$54.73_{0.14}$

Table 5: Quality filter methods and their impact on semantic diversity. The results represent averages over 10 attempts, with standard deviations noted as subscripts.

4 Related Work

Quality Filtering. Pretraining data for large language models often includes low-quality content, such as harmful machine-generated spam or anomalous formats. To filter this data, researchers typically score documents using linear classifiers or language models, then filter based on these scores. High-quality data proxies like Wikipedia, books, and OpenWebText are commonly used. Early studies, such as (Brown et al., 2020; Chowdhery et al., 2023), employed linear binary classifiers, comparing curated datasets to unfiltered CommonCrawl data and used noisy thresholding with Pareto randomness, which, while potentially enhancing diversity, might reduce data quality as suggested by (Xie et al., 2023). Recent studies, such as (Touvron et al., 2023), use Wikipedia as the sole positive class and apply hard thresholding, potentially limiting corpus diversity and introducing biases. Another approach involves language models. For instance, (Wenzek et al., 2019) trained an n-gram model on Wikipedia and categorized documents by perplexity into head, middle, and tail, with low-perplexity documents retained for pre-training (Computer, 2023). Other studies (Wenzek et al., 2019; Soldaini et al., 2023) keep all data to preserve diverse writing styles. Similarly, some filter data based on a model's perplexity, which might bias towards easily predicted texts and discard challenging but high-quality content (Marion et al., 2023). Our approach introduces a quality factor derived from two language models to address this issue.

Scaling Laws. Scaling laws quantify how model size, dataset size, compute budget, and performance relate during neural network training. Initial studies (Hestness et al., 2017; Kaplan et al., 2020) identified a power-law relationship among these factors. (Hoffmann et al., 2022) introduced a unified formula for scaling laws that incorporates data-dependent scaling terms. Recent studies show variations in scaling laws across multilingual (Conneau et al., 2019) and multimodal (Henighan et al., 2020; Cherti et al., 2023) settings. (Aghajanyan et al., 2023) meticulously analyzed this subject, deriving varied scaling law parameters for uni-modal and mixed-modal contexts, highlighting the significant impact of data modality on scaling behaviors. This discovery suggests a hypothesis that varying data quality influences scaling behaviors. A recent study on large language model scaling laws (Bi et al., 2024) confirms data quality impacts both model and data scaling exponents in scaling laws. This paper demonstrates the link between scaling law parameters and data quality, facilitating the selection of high-quality samples based on their scaling attributes.

5 Conclusion

We have presented ScalingFilter for data quality filtering in a reference-free manner. Starting from the scaling law, we demonstrate that the perplexity difference across agnate models of different sizes (i.e. *meta-models*) correlates with data quality positively. We select samples with higher perplexity difference (i.e. *quality factors*) to form the pretraining dataset. By eliminating the bias brought

by reference datasets, our method achieves better downstream performance over several strong baselines while preserving data diversity.

Limitations

There are still some limitations of ScalingFilter that need to be addressed. First, it relies on perplexity difference between two LLMs, which may miss nuanced aspects of text quality like factual accuracy or bias like race bias, social class bias and gender bias, etc. Second, it requires significant computational resources to compute perplexity differences for a large dataset. Third, its applicability to other languages and data-limited domains is uncertain. Future research should address these limitations and further explore the relationship between semantic diversity and model performance, particularly regarding fairness and bias.

Acknowledgments

We sincerely thank Jingcheng Hu and Zheng Zhang for useful discussions. This work was supported by the National Natural Science Foundation of China (No. 62121002, U20B2047), Anhui Provincial Science and Technology Major Project (No. 2023z020006) and the Fundamental Research Funds for the Central Universities.

References

Armen Aghajanyan, Lili Yu, Alexis Conneau, Wei-Ning Hsu, Karen Hambardzumyan, Susan Zhang, Stephen Roller, Naman Goyal, Omer Levy, and Luke Zettlemoyer. 2023. Scaling laws for generative mixed-modal language models. *arXiv preprint arXiv:2301.03728*.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

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

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. 2023. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

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. *arXiv preprint arXiv:1803.05457*.

Together Computer. 2023. Redpajama v1: An open source recipe to reproduce llama training dataset.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Nolan Dey, Gurpreet Gosal, Hemant Khachane, William Marshall, Ribhu Pathria, Marvin Tom, Joel Hestness, et al. 2023. Cerebras-gpt: Open compute-optimal language models trained on the cerebras wafer-scale cluster. *arXiv preprint arXiv:2304.03208*.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR.

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

Dan Friedman and Adji Bousso Dieng. 2022. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv 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.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please: A generic news crawler and extractor. In *Proceedings of the 15th International Symposium of Information Science*, pages 218–223.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. 2020. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. 2017. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

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. *arXiv preprint arXiv:2001.08361*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. 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. *arXiv 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. *arXiv preprint arXiv:1809.02789*.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Houwen Peng, Kan Wu, Yixuan Wei, Guoshuai Zhao, Yuxiang Yang, Ze Liu, Yifan Xiong, Ziyue Yang, Bolin Ni, Jingcheng Hu, et al. 2023. Fp8-lm: Training fp8 large language models. *arXiv preprint arXiv:2310.18313*.

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

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Pete Walsh,

Hannaneh Hajishirzi, Noah A. Smith, Luke Zettlemoyer, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2023. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv preprint*.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2019. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*.

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

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. 2023. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems (NeurIPS)*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

A Appendix

A.1 Hyperparameters of training language models

We train decoder-only transformer (Vaswani et al., 2017) models using Megatron-DeepSpeed (Smith et al., 2022; Shoeybi et al., 2019). The hyperparameters used in the training process is listed in Table A.1.

n params	1.3B
n layers	24
d model	2048
n heads	32
d head	64
Sequence length	2048
Global batch size	256
LR schedule	cosine
Learning rate	2.5×10^{-4}
Min LR	2.5×10^{-5}
Weight decay	0.1
Optimizer	Adam
Adam β_1	0.9
Adam β_2	0.95
Adam ϵ	1.0×10^{-8}
Tokenizer	cl100k_base

Table A.1: Hyperparameters of training language models.

A.2 Derivation of ScalingFilter

A.2.1 Positive correlation between a and the negative tangent slope

Let's start with the parametric loss function introduced by Chinchilla (Hoffmann et al., 2022) scaling law.

$$\hat{L}(N, D) = E + \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}} \tag{8}$$

where E represents the minimal achievable loss, corresponding to the entropy of natural text. The scaling law, indicating the optimized numbers of N and D, follows a power-law form:

$$\begin{gathered} N_{opt} \propto C^{a} \quad D_{opt} \propto C^{b} \\ \text{where} \quad a = \frac{\beta}{\alpha+\beta}, \quad b = \frac{\alpha}{\alpha+\beta} \end{gathered} \tag{9}$$

where a and b represent the model and data scaling exponents, respectively. In order to present α and β with scaling exponents, we have

$$\frac{\alpha}{\beta} = \frac{1}{a} - 1 \tag{10}$$

Let $\eta \doteq \alpha + \beta$, the parametric loss $\hat{L}$ can be presented as:

$$\hat{L}(N, D) = E + \frac{A}{N^{(1-a)\eta}} + \frac{B}{D^{a\eta}} \tag{11}$$

Then, we can obtain the partial derivatives of $\hat{L}$ with respect to N expressed in terms of a and b.

$$\frac{\partial \hat{L}}{\partial N} = A \cdot (a-1)\eta \cdot N^{(a-1)\eta-1} \tag{12}$$

It's obvious that

$$\frac{\partial \hat{L}}{\partial N} < 0 \tag{13}$$

Table A.2: Ablations on hyperparameters used in training 1.3B language models. Numbers in gray represent the default values, as shown in Table A.1.
Abbreviations: BS. = Batch Size, Hella. = HellaSwag, Winog. = Winogrande.

Learning Rate	Global BS.	Hella.	LAMBADA	Winog.	PIQA	ARC	OpenbookQA	BoolQ	Avg
Binary Classification									
2.5×10^{-4}	256	48.13	**48.96**	54.30	69.75	41.66	30.40	61.38	50.65
2.5×10^{-4}	512	46.63	47.72	53.51	68.34	40.81	31.20	59.85	49.72
5.0×10^{-4}	256	49.22	48.71	54.62	70.40	42.45	**33.20**	54.83	50.49
ScalingFilter									
2.5×10^{-4}	256	49.07	48.42	55.09	**70.57**	42.67	31.40	**61.68**	51.27
2.5×10^{-4}	512	46.51	46.24	52.33	69.37	**44.84**	30.00	61.56	50.12
5.0×10^{-4}	256	**49.29**	48.15	**57.06**	70.24	42.95	32.80	60.73	**51.60**

which means that the expected loss decreases when model size increases under same training tokens.

We can further get

$$\begin{aligned}\frac{\partial^2 L}{\partial a \partial N} &= A \cdot \eta \cdot N^{(a-1)\eta-1} \\ &\quad + A \cdot (a-1)\eta \cdot \eta \cdot \ln N \cdot N^{(a-1)\eta-1} \\ &= A \cdot \eta \cdot N^{(a-1)\eta-1} \\ &\quad \cdot [1 + (a-1)\eta \cdot \ln N]\end{aligned} \tag{14}$$

Because $A, N, \alpha, \beta, \eta > 0$, we have

$$A \cdot \eta \cdot N^{(a-1)\eta-1} > 0 \tag{15}$$

and since $1 > a > 0, \eta > 0, N \gg 1$, we have

$$1 + (a-1)\eta \cdot \ln N < 0 \tag{16}$$

Thus, we have

$$\frac{\partial^2 L}{\partial a \partial N} < 0 \tag{17}$$

That means that at a specific N_0, the slope of the tangent to the $\hat{L} - N$ curve ($\frac{\partial \hat{L}}{\partial N}$) decreases as a increases (i.e., the larger the a, the steeper the tangent). In all, we've proven that

$$\frac{\partial(-\hat{L})}{\partial N} > 0, \quad \frac{\partial^2(-\hat{L})}{\partial a \partial N} > 0 \tag{18}$$

Owing to this monotonic relationship, we can infer the value of a from the slope of the tangent. For a given N_0, a steeper tangent (with a greater absolute value of the slope) indicates a larger a:

$$a \propto -\frac{\partial L}{\partial N}\bigg|_{N=N_0} \tag{19}$$

A.2.2 Generalizing from tangent slope to secant slope

It's impossible to calculate the slope of the tangent $\frac{\partial L}{\partial N}$ in the real scenario, we can only acquire the slope of the secant line by assessing the cross-entropy loss on two models with different sizes (i.e. a pair of *meta-models*). Next, we will prove that the slope of the tangent has a positive relationship with the slope of the secant line. Therefore, we can build direct relationship between the slope of the secant line and a.

Given a pair of meta-models with N_p and N_q parameters where $N_p < N_q$, we can denote the slope of the secant line as:

$$\frac{\Delta \hat{L}}{\Delta N} = \frac{\hat{L}(N_q) - \hat{L}(N_p)}{N_q - N_p} = \frac{\int_{N_p}^{N_q} \frac{\partial L}{\partial N} dN}{N_q - N_p} \tag{20}$$

For a larger a, $\frac{\partial L}{\partial N}\big|_{N=N_0} < 0$ is smaller for every N_0. This lead to a smaller $\frac{\Delta \hat{L}}{\Delta N}$, or a larger $-\frac{\Delta \hat{L}}{\Delta N}$, that is

$$a \propto -\frac{\Delta \hat{L}}{\Delta N} \tag{21}$$

A.3 Sampling *vs.* top-k selection

Previous works (Brown et al., 2020; Gao et al., 2020; Xie et al., 2023; Wettig et al., 2024) typically use sampling without replacement, selecting data based on a rating score to balance quality and diversity. This approach often results in improved downstream performance. We conducted experiments to determine whether this sampling strategy could enhance the downstream performance of ScalingFilter. Following (Wettig et al., 2024), we introduced a temperature term τ to adjust sample diversity. Here, $\tau \to 0$ means top-k selection,

Table A.3: Ablations on sampling vs. top-k selection. Note that the top-k results are identical to the original ScalingFilter results reported in Table 1. We compare top-k data selection to sampling without replacement following (Xie et al., 2023; Wettig et al., 2024).

Sampling Method	Hellaswag	LAMBADA	Winogrande	PIQA	ARC	OpenbookQA	BoolQ	Avg
$\tau = 0$ (Top-k)	**49.07**	**48.42**	**55.09**	**70.57**	**42.67**	31.40	**61.68**	**51.27**
$\tau = 1$	47.99	45.93	53.91	68.50	41.46	31.80	61.13	50.10
$\tau = 2$	46.93	48.34	54.06	69.75	41.01	**32.60**	60.28	50.42
$\tau = 3$	47.14	**48.42**	54.46	70.02	42.28	32.00	59.82	50.59

while $\tau \to \infty$ indicates uniform sampling. Results in Table A.3 indicate that top-k selection is the optimal data selection method for ScalingFilter due to its reference-free nature and the unnecessary use of noisy sampling strategies to enhance data diversity.

A.4 Ablation Study on Hyperparameters

To further validate the robustness of ScalingFilter, we conducted ablation experiments using various training hyperparameters on 1.3B models. Our focus was primarily on two hyperparameters: learning rate (default 2.5×10^{-4}) and global batch size (default 256). We doubled the default values for each in the ablation study. The results are presented in Table A.2. The results indicate that an increase in global batch size significantly reduces performance in both settings with different quality filters, as it halves the training steps. Conversely, increasing the learning rate slightly affects downstream accuracy. Overall, ScalingFilter remains robust across a range of training hyperparameters, consistently surpassing binary classification, its top competitor, as shown in Table 1.

A.5 Computational overhead of ScalingFilter

We measured the time required for various quality filtering methods on 360,000 data entries on a machine equipped with an AMD EPYC 7V13 CPU and a single NVIDIA RTX A6000 48GB GPU. The results are summarized in Table A.4.

Method	**Time Required**
ScalingFilter	6 hours
Perplexity Gating	5 hours
Binary Classification	2 minutes

Table A.4: Time required for various quality filtering methods on 360,000 data entries.

Although the ScalingFilter method involves a greater computational overhead, it achieves higher downstream performance and data diversity compared to other methods. The model-based data filtering approach is increasingly being adopted by the community (e.g., LLaMA 3 (Dubey et al., 2024), using LLaMA 2 to perform quality filtering), and model-based methods like ScalingFilter enable iteratively improving data quality through the training of better meta-models.

There might be methods to reduce the computational overhead of ScalingFilter, such as collecting a certain amount of text data with their corresponding quality factors and training a scorer to directly predict the quality factor for any text data. We will leave this type of exploration for future work.

A.6 Qualitative Results and Analysis vs. Perplexity Gating

There are two potential reasons why our proposed ScalingFilter method can provide more complex and diverse data than the Perplexity Gating (Marion et al., 2023; Wenzek et al., 2019) method which use a single model to perform quality filtering.

First, perplexity correlates with the *similarity* between the text and the model's training data, which introduces biases. For example, a model trained on code data will show higher perplexity when predicting literary texts. This was discussed in Section 1, highlighting the sensitivity of perplexity to the alignment between the targeted data and the training data distribution.

Second, perplexity reflects the inherent *complexity* of the text. As noted in Section 1, repeated words like "word word word..." result in very low perplexity because such patterns are easier for language models to predict. While complexity is a potential indicator of data quality, it is not always directly related. Therefore, we aim to minimize the interference brought by complexity when performing data quality filtering.

In contrast, our ScalingFilter method employs a pair of meta-models trained on the same data but with different model sizes. The perplexity computed by each model inherently reflects both the *similarity* to the training data and the *complexity* of the text. By comparing the differences in the perplexities of the two models, we effectively re-

No.	Text
(a)	*Contains disordered formatting:* Two interesting finance and investment posts by David Merkel at The Aleph Blog.....\nExcerpt from Post the First:\nThe main thing to understand here is that the government is not here to help you, but to milk you. The government does not care about you. It cares about its survival. If it can't get sufficient taxes out of the populace, it will use its financing arm, the central bank, to lend to it at preferential rates, while passing on losses to the populace via inflation.\nExcerpt from Post the Second:\nThe main idea is this: buy companies with better prospects than those being sold.\nLabels: Bloggery, Debts, Finance, government, Ideas, Inflation, investing\nSeeing Eye People.....\nHolding on.....\nOpening paragraphs.....\nIke.....\nNumbers.....\nFifty years ago.....\nHappiness.....\nRisk.....\nA gentleman.....\nGone.....\nOpening paragraphs.....\n[TRUNCATED]
(b)	*Contains incomplete sentences:* 「Speaking Out」 Archives\n2019, July Archives\n【#605】Japan Should Seek to Create World Whale-Raising Organization\nJuly 8, 2019 Japan resumed commercial whaling in its 200-mile exclusive economic zone on July 1 after its notification of the withdrawal from the International Whaling Commission on De...\n【#604】Japan Should Wake Up to "Third Black Ship Arrival"\nJuly 1, 2019 "Japan's fate was changed by the United States twice in history - the arrival of Commodore Matthew Perry's black ship in 1853 and Japan's defeat by the U.S. in World War I... [TRUNCATED]

Table A.5: Examples of texts discarded by ScalingFilter but retained by Perplexity Gating.

duce the influence of these factors, thus mitigating the training data bias. Consequently, ScalingFilter enhances both data diversity and quality, as demonstrated in Tables 1 and 5.

To empirically support this, we provide concrete examples of texts that were discarded by ScalingFilter but retained by the single-model Perplexity Gating method. These examples highlight the ability of ScalingFilter to filter out low-quality data that would otherwise be retained due to the biases inherent in Perplexity Gating. Table A.5 presents these examples.

In Example (a), while the first part of the document contains normal text, the second part consists of repeated characters ("....."), which artificially lowers the overall perplexity by reducing the complexity of the latter section. As a result, this document would be retained by Perplexity Gating. However, this document is clearly of low quality, as it contains repeated patterns and poor formatting. Fortunately, ScalingFilter accurately discards this document by eliminating the bias introduced by low complexity.

From the above examples, we can derive an intuitive theoretical explanation. In Example (a), the text exhibits low complexity, leading to low perplexity values across both the large and small meta-models. If we were to directly use the perplexity score for quality filtering, this would introduce a *complexity bias*, as low complexity is not directly indicative of data quality. By calculating the difference in perplexity between the two models, ScalingFilter effectively cancels out the bias caused by complexity, yielding a quality factor that is no longer affected by text complexity alone.