Improving LLM Pretraining by Filtering Out Advertisements

Anonymous ACL submission

Abstract

Data has been recognized as a vital factor for Large Language Models (LLMs), prompting the development of various data selection methods to optimize pretraining data. Among these, the loss-based filtering method has gained popularity due to its straightforwardness. However, our empirical findings suggest that this approach may lead to performance degradation on knowledge-intensive benchmarks, such as the MMLU. To address this issue, we propose filtering out low-information text, particularly advertisements, which constitute a significant portion of internet content. We employed a 100M parameter proxy model to compare these two methods. Despite its smaller size, the proxy model's results accurately predict the downstream metrics when scaled to 3B models. This study demonstrates that a 100M parameter proxy model is sufficient for comparing different data selection strategies, and our experiments across various benchmarks confirm the effectiveness of eliminating advertisements from pretraining data.

1 Introduction

Pre-training on extensive unlabeled and uncrated corpus sourced from internet snapshots (Gao et al., 2020; Penedo et al., 2023; Computer, 2023; Soldaini et al., 2024), empowers large language models (LLMs) with unprecedented capabilities across various domains. Meanwhile, the performance of LLMs scales as a power law with regards as to the data quantity (Kaplan et al., 2020). However, alongside quantity, the quality of the corpus is equally crucial. Recent consensus suggests that high-quality corpora have the potential to significantly alter scaling laws (Sorscher et al., 2022; Hoffmann et al., 2022), enabling performance on par with large-scale models while requiring leaner training costs (Gunasekar et al., 2023; Eldan and Li, 2023)

Therefore, many studies have explored LLM pretraining data selection, including rule-based (Rae et al., 2021), metric-based (Coleman et al., 2019; Marion et al., 2023; Tirumala et al., 2023), gradient-based (Xia et al., 2024) and semantics-based (Brown et al., 2020), each employing different criteria for data quality. Yet, these methods are commonly evaluated by overall metrics, overlooking the detailed influence on different downstream task performances.

Motivated by this gap, we investigate the impact of these strategies on downstream tasks. Surprisingly, our experiments reveal while loss filtering (Marion et al., 2023) enhances text fluency, it can also diminish performance on knowledge-intensive benchmarks like MMLU (Hendrycks et al., 2020). This decline is linked to two main issues: first, the tendency of loss filtering to preferentially preserve fluency-centric marketing content, leading to its overrepresentation; second, the potential exclusion of knowledge-dense texts that incur higher losses when they elude the capturing capabilities of the underlying LLM. Moreover, domain-specific filtering(e.g., Wikipedia classifier (Brown et al., 2020)), although intended to curate domain-relevant data, risks losing valuable cross-domain information.

Based on the previous discussion, we pose two questions:

1. ***Is it possible to devise a data selection strategy that minimizes the inclusion of low-information content while preserving high-information content?***

2. ***How can we quickly assess the effectiveness of data selection strategies in pre-training scenarios?***

To answer the first question, we focus on identifying common traits within web datasets to address the prevalence of low-information content in corpora. Our investigation reveals that advertisements significantly contribute to this issue. In response, we develop an ad classifier, a step beyond the initial mentions in prior work (Wu et al., 2021), providing a detailed approach and thorough analysis of its positive impact on LLM benchmarks, especially knowledge-intensive benchmarks.

To answer the second question, setting aside the costly approach of directly training an LLM end-to-end, D4 (Tirumala et al., 2023) has taken a step forward by exploring the use of proxy metrics from smaller models to validate the quality of pre-training data filtering. However, there are several limitations to these approaches. Firstly, insufficient training (e.g., 1.3B-parameter models on 40B tokens and 6.7B-parameter models on 100B tokens) obscures the manifestation of higher-order abilities, such as knowledge comprehension, as measured by tasks like the MMLU. Secondly, proxy indicators, including perplexity (PPL) from pre-training and various NLP task validation sets, lack sufficient correlation with downstream task performance, limiting domain-specific insights. To address these issues, on the one hand, we evaluate base models after supervised fine-tuning (SFT), which reveals higher-order skills like knowledge comprehension even with limited training. On the other hand, we enhance the proxy indicators for small models by including PPL based on validation sets converted from downstream tasks, enabling early downstream performance predictions and quantifying the correlation between small model proxies and post-SFT large-model downstream metrics. Specifically, we find that the performance of a larger-scale SFT model can be well characterized through the PPL of a 100M proxy LLM on the validation sets.

Using a 100M-parameter proxy model for rapid pre-training iterations (pretraining budget analysis see Section A.5.3), we comprehensively assess popular LLM data selection methods, comparing them against our ad classifier's performance. As depicted in Figure 1, our analysis pipeline highlights the impact of various strategies on model efficacy. Our findings suggest that eliminating advertisement content not only improves performance on knowledge-intensive benchmarks but also yields commendable results across various other capability dimensions within benchmark (see Figure 2).

In summary, our contributions are as follows:

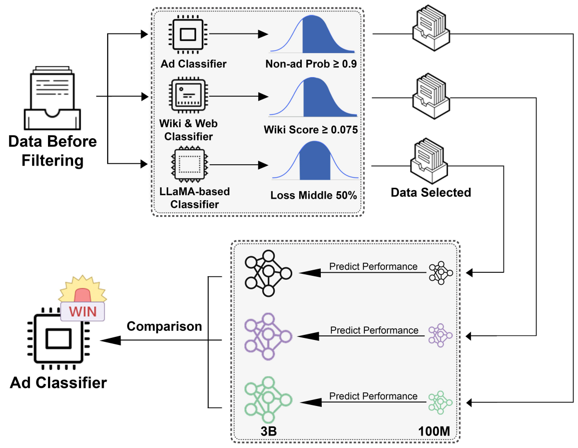


Figure 1: Ad Filtering Outperforms Other Methods Across Three Pre-training Data Selection Techniques

1. We demonstrate that employing a 100M-parameter LLM can reliably predict the utility of pretraining corpora for larger models. We comprehensively establish the correlation between the proxy indicators of the small model and the downstream task metrics of the large SFT model.
2. We emphasize that by using the small surrogate model evaluation mechanism with 100M parameters, we can dramatically reduce the iteration cycles of pre-training data selection strategies, resulting in a substantial budgetary saving of 92.7% (More Cost Analysis see Section A.5.3).
3. We highlight that eliminating advertisement content substantially not only enhances the efficacy of knowledge-intensive benchmarks but also yields commendable results across various other capability dimensions within benchmarks. Additionally, the extent of these performance enhancements varies depending on the data filtering applied, indicating differential downstream effects.

2 Related Work

2.1 Data Selection

As previously emphasized, the importance of high-quality data for training LLMs cannot be overstated. Research on data selection extends across various fields, sharing fundamental principles despite diverse applications. We identify four primary data selection methodologies and provide a systematic analysis of each in the following sections.

Metric-Based Data Selection This line of work primarily focuses on filtering data based on automated metrics generated through dynamic model training. One part of these works explores data filtering on computer vision (CV), with filtering

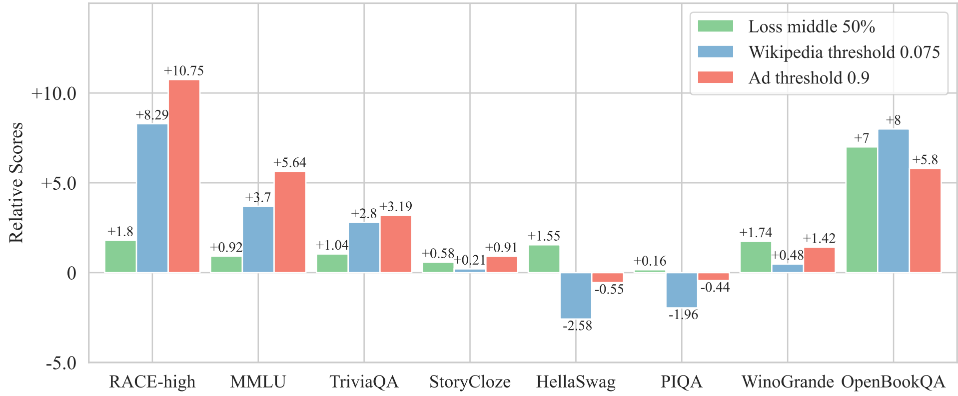


Figure 2: The relative score of performance between different data selection methods with Non-pruning method. In this Figure, each of the models is pre-trained with 300B tokens. See Table 7 for absolute performance of downstream tasks.

strategies including prioritizing hard sample sampling(Coleman et al., 2019), moderate sample sampling(Xia et al., 2023), uncertainty sampling, and filtering based on dynamic changes in statistical values across different epochs(Paul et al., 2021). Another part of the work explores data filtering in the context of NLP and LLM scenarios. The filtering approaches include using perplexity scoring(Marion et al., 2023; Wang et al., 2023), custom IFD(Li et al., 2023a), and multi-metric loss fitting(Cao et al., 2023). In summary, these efforts primarily rely on statistical patterns in the data to obtain valuable samples for model training. However, they struggle to perceive the semantic information in the samples and have difficulty understanding the diversity distribution of the samples.

Semantics-based Data Selection This line of work primarily involves scoring data based on the Wikipedia & Web classifier(Brown et al., 2020; Touvron et al., 2023), reward model(Du et al., 2023), and LLM(Eldan and Li, 2023; Chen et al., 2023; Li et al., 2023b; Sachdeva et al., 2024; Wettig et al., 2024). Intuitively, a semantics-based scoring strategy should have the ability to recognize semantics. However, special attention must be paid to whether the filtering is biased(Gao, 2021).

Geometry-based Data Selection This line of work primarily involves conducting diversity-prioritized sampling based on clustering situations in the feature space and combines with metric-based or semantic-based strategies(Maharana et al., 2023; Du et al., 2023; Tirumala et al., 2023).

Gradient-based Data Selection This line of research leverages Influence Functions(Xia et al., 2024; Engstrom et al., 2024; Yu et al., 2024; Koh and Liang, 2017; Ling, 1984; Grosse et al., 2023; Schioppa et al., 2022) to identify training data points that exert the most significant impact on the validation points. Concurrent studies like LESS, DsDM, and MATES have investigated high-cost influence data selection in LLMs from multiple angles, such as the Adam optimizer, data models, and evolving data influences. These methods, however, depend on a validation set to assess the impact of training data. Thus, constructing a robust validation set and preventing overfitting during the selection process for downstream tasks are critical considerations.

2.2 Evaluation of Pre-training Data Selection

In addition to D4 (Tirumala et al., 2023) as mentioned in section 1, (Marion et al., 2023) exhibits pre-trained models of 124M and 1.5B parameters with validation set perplexity and downstream SFT task evaluation. However, it is limited by the use of a validation set whose domain is aligned with the training dataset's distribution. Perplexity rankings within in-domain validation sets can be inconsistent across different data selection strategies, potentially misrepresenting a model's true capabilities. Furthermore, it only reports classification task performance on GLUE after SFT, offering a partial view of LLM's overall abilities. We not only extend beyond those mentioned in comparison with D4 but also include our choice of validation sets. We select three types of validation sets, which are all out of training set domains, to reflect the model's generalization on smaller scales.

3 Method

As previously outlined, the data selection pipeline is depicted in Figure 1. Within this pipeline, a small proxy model evaluation mechanism is employed to predict the downstream performance of the larger SFT models. Our investigation commences with an analysis of prevalent LLM data selection techniques, including the loss filter and the Wikipedia Classifier, with a focus on their influence on downstream tasks. Subsequently, we delve into the development and efficacy of the advertisement classifier. The critical components of this process are elucidated below.

3.1 Small Surrogate Model Evaluation Mechanism

The quintessence of our proposed Small Surrogate Model Evaluation Mechanism is to establish a correlation between the performance of small models and the downstream task metrics of larger models. This allows the performance of smaller models to predict the downstream task performance of larger models, thereby significantly reducing the iterative costs associated with pretraining data selection methods. To rigorously analyze the efficacy of our proposed Evaluation Mechanism, please refer to Figure 3(b) for an illustrative depiction of the overall process. detailed terms definitions and process descriptions see the Appendix A.1.1.

After substantiating the effectiveness of the overall framework, the process can be streamlined for practical application, as shown in Figure 3(a). For any two data selection schemes, it is sufficient to compare the Surrogate Indicators on the Surrogate Model to determine the superior data selection strategy. This approach can significantly lower the iterative costs associated with pretraining data selection methods.

Our intuitive understanding of the proposed mechanism is derived from the theoretical analysis presented in (Hoffmann et al., 2022), which suggests that even with identical training computation, different combinations of model size and data size can lead to varying pretraining losses. Consequently, a logical approach is to control for the pretraining model size and hyperparameters and then observe the validation set losses (equivalent to PPL) of models pre-trained with different data combinations on a high-quality, diverse validation set that is strongly relevant to downstream tasks. This allows for the assessment of the pretraining efficacy of LLMs. Building on this theory, it is also intuitive to use the pretraining performance of smaller models (indicated by PPL) as a surrogate to predict the pretraining capabilities of larger models under the same data conditions, with the downstream task performance as the metric of evaluation. Our proposed mechanism significantly differs from the deep learning core-set data selection via proxy as described in (Coleman et al., 2019). Detailed analysis can be seen in Appendix A.5.1

We summarize contributions of Small Surrogate Model Evaluation Mechanism in Appendix A.5.2.

3.2 Advertisement Classifier

In our examination of the English Common Crawl corpus, we observe a significant prevalence of marketing content and product placements. Notably, product placements frequently exhibit redundancy and lack of fluency, whereas marketing content is typically distinguished by its high fluency. Given this background, we aim to sift through the data, removing ads to potentially enhance the corpus with knowledge-intensive material of higher quality for LLM pretraining. We filter out advertisements through a well-designed ad classification process, involving data sampling from RefinedWeb, human annotation, and a binary BERT model to distinguish non-ads from ads. The process was iterative, with continuous manual review and re-labeling of misclassified samples until achieving a desired low ad misclassification rate. The development of this ad classifier, aligned with human judgment, is depicted in Figure 4 and detailed ad classifier construction process can be seen in Appendix A.1.3

Unlike Yuan1.0, which uses a ternary classifier to filter a Chinese corpus into low-quality, advertising, or high-quality texts based on repetition rates (Wu et al., 2021), we categorize texts as advertising or non-advertising by focusing on promotional content and product placement. Yuan1.0's methodology, which targets coherent but redundant texts like website descriptions, differs from our content and style-based approach. Furthermore, while Yuan 1.0 has not disclosed their pre-training experiment results, we have detailed ours in A.4.3.

3.3 Baselines

We compare advertisement classifier with several baselines. The comparative experiments are conducted under the same sequence of data points.

None-Filter: This means using all data points during the training process.

Wikipedia and Web Classifier: This method utilizes a binary classifier to distinguish between high-quality, knowledge-rich content from Wikipedia

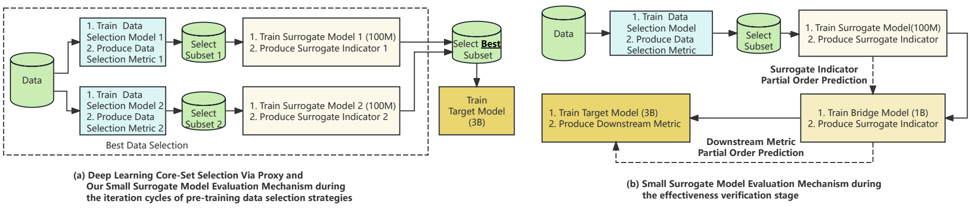


Figure 3: Small Surrogate Model Evaluation Mechanism during the iteration cycles of pre-training data selection strategies and during the effectiveness verification stage

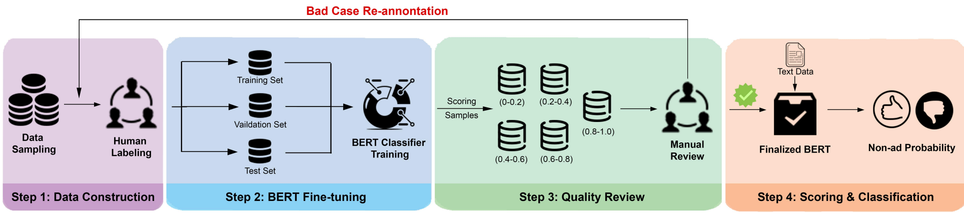


Figure 4: Pipeline of Data Labeling BERT Classifier Training

and lower-quality text extracted from the Common Crawl dataset (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023). We provide more details in Appendix A.1.5

Loss Filter: This filtering technique employs pre-trained models to compute the perplexity of texts across the dataset and then uses perplexity to filter data (Marion et al., 2023; Xia et al., 2023). We provide more details in Appendix A.1.4

LESS: LESS (Xia et al., 2024) selects training samples that have a significant impact on validation data points. Due to the high computational cost associated with LESS, the size of the pretraining dataset and the scale of the pretraining model are reduced to manage expenses. Essential comparative experiments are conducted to compare LESS and advertisement classifier. We provide more details in Appendix A.1.6

4 Experiments

4.1 Training Details

Our pretrain experiments are conducted with the RefinedWeb dataset (Penedo et al., 2023), which uses advanced rule-based filtering and deduplication methods, **without any secondary classifier-based filtering**. In this way, we are able to implement detailed ablation studies, comparing the impacts of various filtering methods. and SFT experiments are with Flan Collection (Longpre et al., 2023). In our experiment, we train decoder-only Transformer from scratch only once for each experiment due to constraints of training costs. We provide full details of pre-training and SFT hyper-parameters in Appendix A.2.1 and A.2.2. Meanwhile, we estimate computational costs in A.2.3.

4.2 Evaluation Metrics

We consider two key metrics for evaluation: validation set PPL and downstream benchmark metrics, with a detailed correlation analysis in Section A.3.1.

Validation Set Perplexity To evaluate the model's impact on downstream tasks, we utilize three distinct validation datasets, with each catering to different domains, to offer an early performance assessment for models with 100M parameters. Detailed descriptions are available in Section A.2.4.

Downstream Benchmark Metrics We select 10 tasks across five categories to gauge our model's effectiveness on downstream tasks: text completion (Mostafazadeh et al., 2017), reading comprehension (Lai et al., 2017), common-sense question answering (Zellers et al., 2019; Bisk et al., 2020; ai2, 2019; Mihaylov et al., 2018), factual question answering (Kwiatkowski et al., 2019; Joshi et al., 2017), and examination(Hendrycks et al., 2020). An overview of these tasks is presented in A.2.5.

5 Result

100M LLM can reliably predict the utility of pretraining corpora for larger models. We quantitatively assess the correlation between the proxy metric (validation set PPL) of the 100M model and the downstream task metrics of the 3B SFT model with a three-phase correlation analysis.

Phase 1: Figure 5 shows a high correlation in PPL between the 100M and 1B models across most validation sets with exceptions noted in specific datasets such as RACE-middle and TrivialQA.

Phase 2: From Figure 6, the PPL of the 1B and 3B models show a significant correlation across most validation datasets with exceptions noted in specific datasets such as RACE-middle and TrivialQA.

Phase 3: From Figure 7, lower PPL in different 3B models on the validation sets correlates with higher downstream task metrics.

More detailed analysis and more figures can be seen in Section A.3.

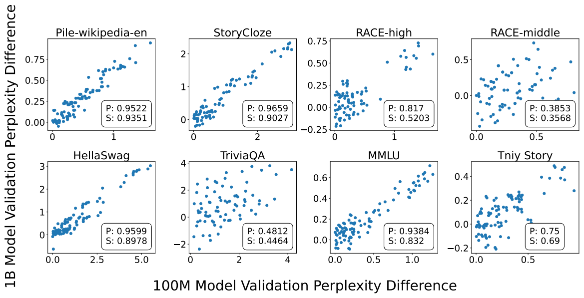


Figure 5: Validation Perplexity Difference Comparison Between 100M and 1B Model

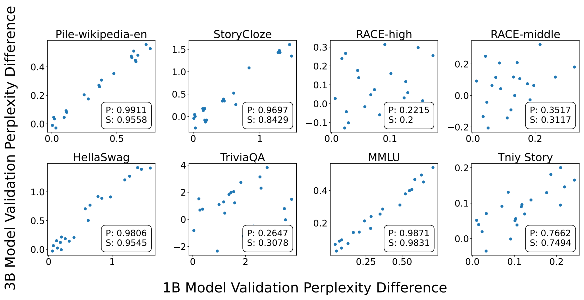


Figure 6: Validation Perplexity Difference Comparison Between 1B and 3B Model

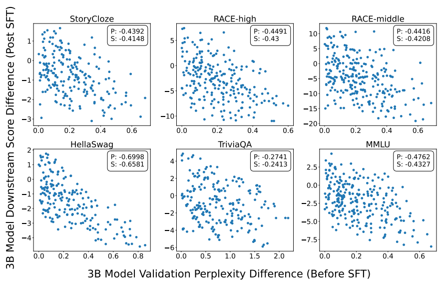


Figure 7: 3B Model Validation Perplexity Difference vs. 3B Model Downstream Score Difference

Using 100M small surrogate model evaluation mechanism, we can dramatically reduce the iteration cycles of determining optimal thresholds and retention for different data filtering strategies.

Table 1 shows the partial order ranking of validation sets at 100M model and 100B token budget with different loss thresholds. Synthesizing these results, we discern a notable decrease in PPL (indicating improved performance) on HellaSwag for PPL@loss middle 50%, a marked increase (indicating decreased performance) on MMLU and Wikipedia-en, and a relatively lower PPL (indicating better performance) on Tiny Story. After comprehensive consideration, we selected the loss middle 50% threshold, which corresponds to a data remaining ratio of 53.9%. More detailed analysis can be seen in Appendix A.4.1.

Table 2 shows the partial order ranking of these validation sets at 100M model and 100B token budget with different Wikipedia classifier thresholds. Synthesizing these findings, we note a significant reduction in PPL (indicating performance improvement) at PPL@thresh0.075 for MMLU and Pile-Wikipedia. For HellaSwag, there is an increase in PPL (indicating worse performance, likely due to the loss of relevant data). In the case of Tiny Story, a PPL@thresh0.25 increases perplexity compared to no filtering, but PPL@thresh0.075 and PPL@thresh0.0255 initially reduce PPL, aligning with unfiltered data. This pattern underscores the nuanced effect of data filtering on text generation fluency. After comprehensive consideration, we selected a threshold of 0.075, with a data remaining ratio of 63.4%. More detailed analysis can be seen in Appendix A.4.2.

Table 3 shows the partial order ranking of these validation sets at 100M model and 100B token budget with different ad classifier thresholds. PPL@threshold 0.95 experiences a significant increase on HellaSwag, indicating a decline in performance. Conversely, PPL@threshold 0.9 maintains a relatively lower score on MMLU, Tiny Story, and Pile-wikipedia-en, which suggests better performance. Moreover, the performance of PPL@threshold 0.9 on HellaSwag shows negligible differences when compared to other thresholds. Consequently, we have selected a threshold of 0.9, with the data retention rate being 64.1%. More detailed analysis can be seen in Appendix A.4.3.

Ad Classifier yields superior performance on most tasks when compared to other methods, especially in knowledge-intensive benchmark MMLU. In other benchmarks, this method also shows commendable results..

We evaluate the performance of these filtering methods, including none filter, loss filter, wikipedia

	None-filtered	loss middle 50%	loss middle 30%
MMLU	0	1	2
HellaSwag	2	0	1
Tiny Story	2	1	0
Pile-wikipedia	0	1	2

Table 1: Validation Perplexities Partial Order Ranking of Different Loss Thresholds at 100B token.(0 means lowerest ppl and 2 means largeest ppl.)

	None-filtered	Threshold 0.025	Threshold 0.075	Threshold 0.25
MMLU	3	1	0	2
HellaSwag	0	1	2	3
Tiny Story	0	0	0	3
Pile-wikipedia	3	2	1	0

Table 2: Validation Perplexities Partial Order Ranking of Different Wikipedia and Web Thresholds at 100B token. (0 means lowerest ppl and 2 means largeest ppl. Same order will show lower order rank)

	None-filtered 100M/1B	Ad 0.9 100M/1B	Wikipedia 0.075 100M/1B	Loss 50% 100M/1B
MMLU	2/2	0/1	0/0	3/3
HellaSwag	1/1	2/2	3/3	0/0
RACE-High	3/2	0/0	2/2	1/0
RACE-middle	3/3	0/0	2/1	1/1
TivialQA	2/3	0/0	1/1	3/2
StoryCloze	2/3	1/1	3/1	0/0
Tiny Story	2/3	0/0	3/1	1/2
Pile-Wikipedia	2/2	1/0	0/0	3/3

Table 4: Validation Perplexities Partial Order Ranking of Different Data Selection Methods with 100M/1B model. (0 means the lowest ppl, and 2 means the largest ppl. The same order will show a lower order rank)

	None-filtered	Ad 0.9	Wikipedia 0.075	Loss 50%
MMLU	3	0	1	2
HellaSwag	1	2	3	0
RACE-High	3	0	0	2
RACE-middle	3	0	0	2
TivialQA	3	0	1	2
StoryCloze	3	0	2	1

Table 5: Downstream Metric Partial Order Ranking of Different Data Selection Methods with 3B model (0 means highest metric and 2 means lowerest metric. Same order will show lower order rank)

classifier, and ad filter, across different model sizes (100M, 1B, and 3B models), with a particular focus on their impact on downstream tasks. The results (validation perplexities partial order ranking of Table 4 and Table 5, downstream benchmark metrcis of Table 7 indicate that ad filter consistently improves performance across most tasks, especially in knowledge-intensive tasks such as the MMLU benchmark. In contrast, loss filter shows moderate performance in knowledge tasks, while wikipedia classifier exhibited negative impacts in benchmarks focused on common sense benchmarks. More detailed analysis and more figures can be seen in Appendix A.4.4 and Appendix A.4.5.

Limited by computational costs, we conduct a focused comparison on RefinedWeb 200B token new shuffle subset, comparing none-filter, ad classifier, and LESS in terms of perplexity rankings at 100M and 1B model scales. From perplexity curves (Figure 8), ad filter is generally lower than LESS across most validation sets, except on the Hellaswag validation set. From Figure 17, Although ad filter exhibits a higher PPL on Hellaswag compared to other methods, the impact on downstream task performance (Table 7) is minimal. Since LESS requires pre-prepared validation sets for calculating influence scores, it may introduce the risk of overfitting downstream tasks. In contrast, the advertisement classifier is constructed without using any validation set information, making it a more universally pre-training data filtering approach.

5.1 Analysis of Data Remaining Ratios for Different Data Filtering Methods

We evaluate the data retention ratios of various filtering strategies on validation sets as an indirect measure of their influence on downstream tasks. Despite the validation set partly originating from downstream instruction tasks, which diverge in format from our pre-training corpus, we consider these tasks as domain-specific corpus material. Consequently, we propose that the varying data remaining ratios across domains within our validation set can provide insights into the impacts of data filtering strategies on these domains. Furthermore, comparing data retention ratios for different strategies within the same validation set domain can yield relative effectiveness insights.

As shown in Table (6), the loss filtering method results in a reduced data remaining ratio on the MMLU, indicating potential negative impacts on the MMLU benchmark. This observation aligns with the finding that loss filtering falls short of

	None-filtered	Threshold 0.4	Threshold 0.6	Threshold 0.8	Threshold 0.9	Threshold 0.95
MMLU	5	4	2	2	1	0
HellaSwag	1	1	0	3	3	5
Tiny Story	5	3	3	2	0	0
Pile-wikipedia	5	3	3	1	1	0

Table 3: Validation Perplexities Partial Order Ranking of Different Ad Thresholds at 100B token.(0 means lowerest ppl and 2 means largeest ppl. same order will show lower order rank)

	Wiki threshold 0.075	loss middle 50%	Ad threshold 0.9
Pile-Wikipedia	68.8%	17.5%	98.3%
StoryCloze	0.1%	63.2%	98.9%
RACE-High	67.6%	75.9%	74.5%
RACE-Middle	45.5%	70.8%	88.4%
HellaSwag	0.3%	52.2%	95.2%
TriviaQA	0.1%	7.2%	99.5%
MMLU	82.7%	11.1%	94.4%
Tiny Story	33.0%	5.0%	99.6%

Table 6: Data Remaining Rates for Different Data Filtering Schemes on Downstream Validation Sets of Different Domains

	Data Remaining	Reading Comprehension	Exam	Factual QA	Text Completion	Common-Sense QA
		RACE-High ǀ RACE-middle	MMLU	Natural Question ǀ TriviaQA	StoryCloze	HellaSwag ǀ PIQA ǀ WinoGrande ǀ OpenBookQA
No Pruning	100%	29.33 ǀ 32.38	29.71	11.19 ǀ 30.61	75.15	64.75 ǀ 77.15 ǀ 57.93 ǀ 22
Loss middle 50%	53.9%	31.13 ǀ 36.84	30.63	9.56 ǀ 31.65	75.73	66.3 ǀ **77.31** ǀ **59.67** ǀ 29
Wikipedia threshold 0.075	63.4%	37.62 ǀ 41.57	33.41	**12.35** ǀ 33.41	75.36	62.17 ǀ 75.19 ǀ 58.41 ǀ 30
Ad threshold 0.9	64.1%	**40.08** ǀ **45.82**	**35.35**	12.08 ǀ **33.8**	**76.06**	64.2 ǀ 76.71 ǀ 59.35 ǀ 27.8

Table 7: The downstream metric of each data selection method, including Reading Comprehension, Exam, and Factual QA, with 3B models pretrained with 300 billion tokens. Underlined results surpass the baseline performance with no pruning. The best results for each task are marked in bold.

other strategies in the 3B SFT-enhanced MMLU context. Similarly, the Wikipedia filtering strategy, with its lower data retention ratio on HellaSwag, suggests a detrimental effect on the common sense benchmark, corroborating its underperformance in post-3B SFT HellaSwag evaluations. Interestingly, the ad filtering strategy consistently exhibits high data remaining ratios across the validation set, an outcome achieved without incorporating any information from the validation set.

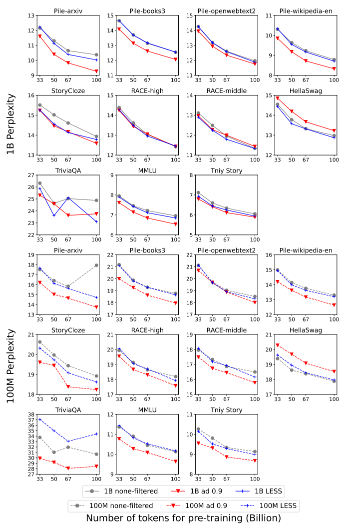


Figure 8: Validation Perplexities Comparison Between 100M & 1B Models between ad filter and LESS

5.2 Analysis about Potential Confounding Factors

We provide data metric visualizations to further analyze potential confounding factors:

1. Does the removal of advertisements affect the distribution of data retention lengths, thereby influencing model performance?

We visualized the length distribution (in bytes per sample) of data retained with ad filter threshold of 0.9 compared to the distribution with none-filter in Figure 18. It is evident that there is no significant change in the distribution of data lengths before and after filtering. This observation effectively rules out the possibility that the length distribution of the data serves as a confounding factor.

2. Does ad removal impact the distribution of data across different thematic domains, thereby influencing model performance?

We perform k-means clustering on the whole dataset, thereby generating 15,000 clusters. All data are assigned to the nearest cluster based on the nearest neighbor distance. We then randomly select 100 samples from each centroid and subjected them to ad classification scoring by LLAMA2-chat, yielding an average ad score for each cluster. Subsequently, we calculated the proportion of data reduction after applying the ad filter threshold of 0.9 for each cluster. We then assessed the consistency between the average ad scores of all clusters and the proportion of data reduction post-filtering.

The results revealed a Pearson Correlation Coefficient of 0.878 and a Spearman Correlation Coefficient of 0.876, indicating that advertisement filtering indeed affects the distribution of data across different thematic domains. Clusters more closely related to advertisement themes experienced greater data reduction. This finding intuitively validates our proposed advertisement data filtering approach, confirming that it effectively employs the factor of advertisement content to refine the dataset, thereby enhancing model performance.

6 Conclusion

Our research demonstrates that using loss metrics for selecting pretraining data can negatively impact performance on complex, knowledge-intensive tasks like MMLU. We improve data quality for LLM pre-training by implementing a specialized ad classifier to eliminate low-information content, enhancing model performance across various benchmarks. Additionally, we introduced a cost-effective and efficient evaluation method by using a smaller LLM as a proxy to forecast the success of larger models. This approach has significantly reduced resource costs by 92.7%, enabling rapid iterations in data selection strategies and offering a scalable, practical solution for future LLM development.

Limitations

Small models to predict the reasoning ability of large models: The reasoning ability of existing LLMs emerges under certain conditions, such as model size, high-quality mixed data, and a certain computational budget. We do not have the time to explore whether it is possible to use smaller models on web datasets with appropriate proxy indicators to reflect the reasoning ability of a medium-sized model. There is no consensus yet on the origins of the reasoning mechanism produced by LLMs. If the changes in reasoning ability could be reflected through proxy indicators on smaller models, it would greatly aid in understanding the origins of reasoning abilities.

Ad filtering in conjunction with other filtering solutions: Ad filtering is about removing corpora with advertising content. Although loss filtering may discard knowledgeable content, it can still eliminate a lot of incoherent corpora. What kind of integrated scheme could complement the advantages of multiple filtering solutions? Limited by time and cost, we have not explored the integration of multiple existing filtering solutions in this work.

7 Ethics Statement

7.1 Data Collection

All the datasets we use in our work are from publicly available resources (RefinedWeb). And we will open part of quality scores of this dataset. The data License will follow RefineWeb.

7.2 Human Labeling

For the BERT advertisement classifier, we curate a dataset of 40,000 samples from RefinedWeb, which are then labeled as either advertisement (ad) or non-advertisement (non-ad) by annotators. Because the annotators are formal employees of the company and are subject to confidentiality requirements regarding their remuneration, it is not possible to provide information on average salaries to the outside. The form and instructions presented to human evaluators are shown in Figure 14.

References

2019. Winogrande: An adversarial winograd schema challenge at scale.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

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.

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

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

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.

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

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

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. `https://github.com/open-compass/opencompass`.

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

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*.

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

Leo Gao. 2021. An empirical exploration in quality filtering of text data. *arXiv preprint arXiv:2109.00698*.

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*.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, et al. 2023. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

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

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

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*.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

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

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

Robert F Ling. 1984. Residuals and influence in regression.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Adyasha Maharana, Prateek Yadav, and Mohit Bansal. 2023. D2 pruning: Message passing for balancing diversity and difficulty in data pruning. *arXiv preprint arXiv:2310.07931*.

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.

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 *EMNLP*.

Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. 2017. Lsdsem 2017 shared task: The story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 46–51.

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

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*.

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*.

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. *arXiv preprint arXiv:2402.09668*.

Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. 2022. Scaling up influence functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8179–8186.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. 2022. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536.

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

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S Morcos. 2023. D4: Improving llm pretraining via document de-duplication and diversification. *arXiv 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, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Yue Wang, Xinrui Wang, Juntao Li, Jinxiong Chang, Qishen Zhang, Zhongyi Liu, Guannan Zhang, and Min Zhang. 2023. Harnessing the power of david against goliath: Exploring instruction data generation without using closed-source models. *arXiv preprint arXiv:2308.12711*.

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

Shaohua Wu, Xudong Zhao, Tong Yu, Rongguo Zhang, Chong Shen, Hongli Liu, Feng Li, Hong Zhu, Jiangang Luo, Liang Xu, and Xuanwei Zhang. 2021. Yuan 1.0: Large-scale pre-trained language model in zero-shot and few-shot learning.

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

Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. 2023. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

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

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 Method Details

A.1.1 Small Surrogate Model Evaluation Mechanism Details

Here, we show the terms involved in Figure 3.

Data Selection Model: The term refers to a model used to filter pre-trained data, which will produce data selection metrics used to filter the data. The ordering of these metric values can filter out the required subset of data.

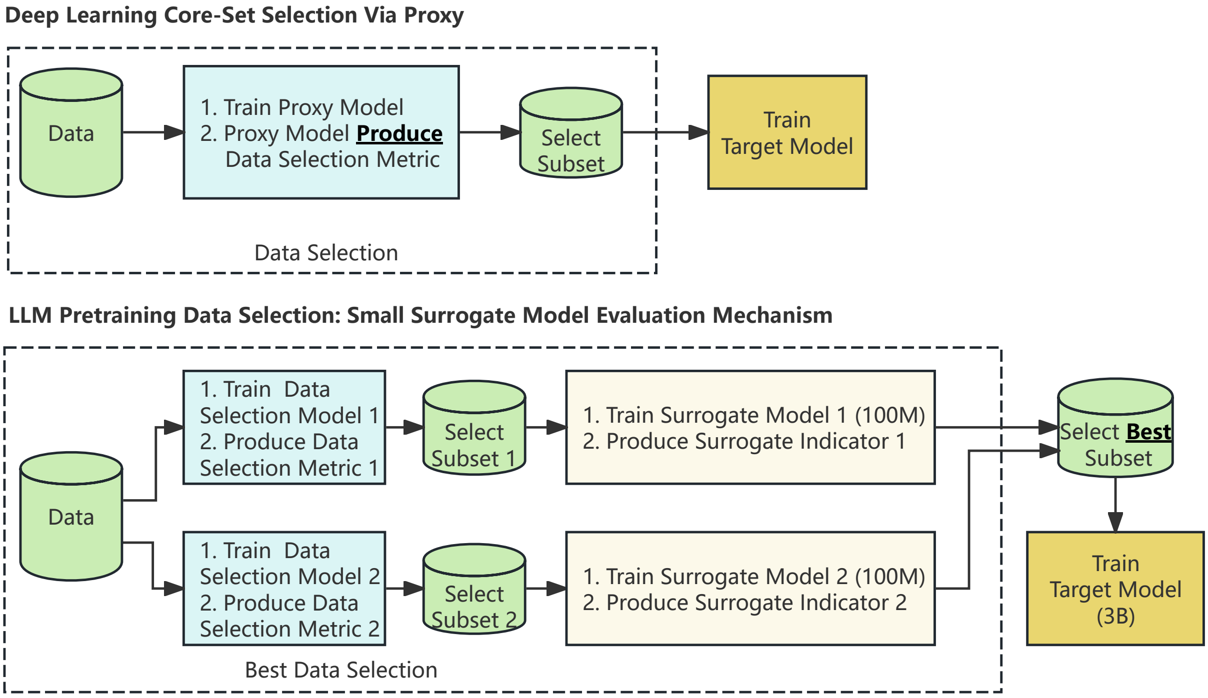
Surrogate Model: This term refers to a surrogate model utilized to validate the effects of pretraining on larger-scale models. The expectation is that the pretraining outcomes on the proxy model will provide early insights into the performance of larger models, thereby significantly reducing the computational cost associated with hyperparameter experiments for data selection strategies. In this study, the proxy model is a 100M model.

Surrogate Indicator (Surrogate Metric): This is a surrogate metric for assessing the pretraining performance on the proxy model. The proxy indicator on the proxy model can predict the target model's performance in downstream tasks. The proxy indicator used in this study is PPL.

Target Model: This term refers to the pretraining model that is the focus of our evaluation. Notably, even when considering smaller-scale models, the application of SFT can significantly reveal the impact of data selection strategies and the model's higher-order capabilities in downstream tasks. Meanwhile, due to computational resource constraints, the target model in this study is specified as a 3B model post-SFT.

Downstream Metrics: These metrics assess the target model's capabilities across various downstream tasks. The tasks encompass 10 different types, with specific descriptions provided in A.2.5.

Bridge Model: This is an intermediary model introduced to enhance the robustness of the transition from the proxy model's proxy indicator to the target model's downstrea metrics. The rationale for introducing a bridge model is the prohibitively high experimental cost of the target model, which precludes exhaustive ablation studies. Hence, the bridge model is employed to conduct as many hyperparameter experiments as computationally feasible to increase the robustness of the correlation analysis. In this study, the bridge model is a 1B model.



(a) Deep Learning Core-Set Selection Via Proxy and Our Small Surrogate Model Evaluation Mechanism during the iteration cycles of pre-training data selection strategies

Figure 9: Small Surrogate Model Evaluation Mechanism vs Deep learning Core-set selection

A.1.2 Ad Classifier

When evaluating the effectiveness of our BERT classifier, we employ a bootstrap method, sampling 1000 times, with each time randomly selecting 50% of the data to calculate precision and recall values at different thresholds. The Precision-Recall curve for BERT training, complete with confidence intervals, is shown in Figure 15, demonstrating our classifier's effectiveness in identifying ads, closely mirroring human judgment.

Furthermore, we try different thresholds(0.4, 0.6, 0.8, 0.9 and 0.95) for our BERT advertising classifier, which outputs a probability of a text being non-ad data. Not only do we include data remaining ratios under these thresholds in Table (8), but we also take the precisions and recalls of ad and non-ad prediction into account so that we can make the best choice for the threshold of ad classification. Detailed experiment result can be found in Appendix A.4.3

A.1.3 Ad classifier Construction Process

In this section we will explain in detail the process of building the advertisement classifier in Figure 4.

1. **Data Sampling**

The core challenge in the data sampling phase is how to select a representative set of advertisement/non-advertisement datasets. Without broad sampling, it's easy to fall into the pitfall of out-of-distribution. There are many thematic sampling schemes available, ranging from traditional NLP techniques like LDA ((Blei et al., 2003)) for unsupervised topic analysis to unsupervised clustering techniques. After completing the theme mining of the pre-training dataset, sampling a batch of samples for each theme can accomplish representative sample sampling.

2. **Human Labeling**

The human labeling operation is divided into two steps: manual labeling (by annotators) and secondary audit detection.

2.1. **Manual Labeling**

Firstly, we establish the categories of advertisements and preliminary identification standards through experts, specifically divided into insert advertisements, full-text marketing advertisements, and soft advertisements. Secondly, to align annotators' perception of advertisements, we deliver a small amount of advertisement data for trial annotation. After reviewing the results, we find significant differences in annotators' perception of soft advertisements, whose definition is indeed vague. There-

Threshold	Non-ad Precision	Non-ad Recall	Ad Precision	Ad Recall	Data Remaining
0	71.4%	100.0%	–	0.0%	100%
0.4	80.0%	96.6%	82.1%	39.7%	88.7%
0.6	86.2%	94.5%	81.8%	62.1%	82.9%
0.8	89.7%	89.7%	74.1%	74.1%	73%
0.9	91.9%	86.2%	70.2%	81.0%	64.1%
0.95	95.1%	80.0%	64.2%	89.7%	55.2%

Table 8: Data Remaining Ratio, Precision and Recall Under Different Non-ad Probability Thresholds

fore, although we require the annotation of soft advertisements, in actual training, soft advertisements are classified as normal samples to avoid classifier confusion due to unclear standards. Thirdly, to improve annotators' efficiency, we also provide an auxiliary labeling feature based on the open-source large model LLAMA2-chat to help annotators better understand the standards of advertisements and enhance the annotation effect. Finally, after aligning the annotators' perception of advertisements, the annotators begin bulk manual annotation.

2.2. **Secondary Audit**

Auditors are responsible for batch sampling quality audits of manually labeled data, sending back batches that do not meet standards and re-labeling, at the meanwhile increasing the frequency of data review for that annotator. The audit continues until the rejection rate drops below a certain threshold.

3. **BERT Fine-tuning**

At this step, we obtain a certain amount of positive and negative sample data (each about 10w); we divide it into a training set and a validation set (same distribution); the test set is specially selected during the labeling process, consisting of representative advertisement and non-advertisement data (each about 1k); Then We train a BERT classifier using manually annotated data with non-ad text to be labeled 1 and ad text to be labeled 0.

4. **Data quality review**

In this step, we apply the high-quality classifier obtained from training to the large-scale pre-training data, obtaining large-scale scoring data through BERT scoring. Furthermore, we conduct quality checks on the data obtained from the large-scale data. The specific operations are as follows: We sample data within different scoring intervals, specifically dividing the classification into 5 buckets, each interval of 0.2 as one bucket, a total of 5 buckets, and perform bucket inspection. During bucket inspection, we prioritize providing diverse samples based on thematic information for auditors to review. When the volume of data that does not meet the audit standards reaches a certain level within a bucket under a certain theme, we will redirect the relevant data in this bucket back to the annotators for labeling, add it to the classifier's training after completing the labeling process, and repeat this process until the inspection is qualified, finally obtaining a high-quality advertisement classifier for the final bulk scoring. This data review process can also be further optimized based on the idea of active learning.

5. **Bert model evaluation**

we apply the trained BERT on another batch of manually annotated data for ad classification to validate the effectiveness of our classifier, where we reach the average precision of 96.63% for non-ad classification and 80.66% for ad classification. The resulting Precision-Recall curve with confidence intervals and data remaining ratios under different thresholds are shown in A.1.2.

6. **Scoring Classification**

After the manual review is completed, the final version of the advertisement classifier is applied to the RefinedWeb dataset to obtain the advertisement score for each sample, which is used for subsequent steps.

A.1.4 Loss Filter

This method leverages pre-trained models to compute perplexity for the entire dataset. It is indicated that employing moderate perplexity thresholds for data filtering can enhance training efficiency (Marion et al., 2023; Xia et al., 2023), a hypothesis we will explore in depth.

We utilize LLaMA2-7B for dataset scoring and adopted a strategy of remaining mid-range data for comparative experiments (Marion et al., 2023). We evaluate the effects of no filtering, remaining the middle 50% of all data based on loss ranking, and retaining the middle 30% of all data based on loss ranking. The respective data remaining ratios for no pruning, loss middle 50%, and loss middle 30% are 100%, 53.9%, and 32%. Detailed experiment result can be found in Appendix A.4.1

A.1.5 Wikipedia and Web Classifier

Contrasting with the ad filter, this strategy employs a binary classifier to separate high-quality, knowledge-rich text (e.g., Wikipedia) from low-quality Common Crawl data (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023). Despite superficial similarities to the ad filter, this method focuses on the automatic segregation of text corpora, aiming to enhance data quality for pre-training. However, defining clear-cut divisions between these text types presents significant challenges and may inadvertently introduce biases.

We employ a quality classifier trained with RedPajama[1]. Although a threshold of 0.25 is recommended to filter out low-quality data, we compare the experimental effects of four sets of thresholds (0, 0.025, 0.075, 0.25). The data remaining rates of no pruning, threshold 0.025, threshold 0.075, and threshold 0.25 are 100%, 78.6%, 63.4%, and 42%. We will delve into a detailed analysis of these biases in subsequent Section A.4.2.

A.1.6 LESS Details

We utilize the open-source code from LESS[2] to filter our pretraining data. Although LESS is originally designed for filtering data for instruction tuning, its methodology can be straightforwardly adapted for pretraining data selection without significant modifications.

We adhere to the training hyperparameters established by LESS, with the only modification being the substitution of the training data with the pre-training data from RefinedWeb. We follow the LESS framework, conducting training on 8 GPUs for 4 epochs, processing a total of 1 billion tokens of pre-train data and producing a LORA LLAMA-7B model. Due to the high cost associated with gradient computation, we restrict our use of the influence score calculation to the checkpoint from the final epoch only.

On the pre-training dataset side, we choose a subset of 200 billion tokens from RefinedWeb. We set a retention rate of 49.3%, thus filtering out 100 billion tokens of pretraining data for experimental comparison. This retention rate is notably close to that used in an advertising filtering scenario, where a retention rate of 60% is typical under a 0.9 filtering threshold, ensuring that the volumes of data retained in both cases are comparably similar.

For validation, we select development sets from various benchmarks, including HellaSwag, MMLU, Pile-Wikipedia, RACE-High, StoryCloze, and Tiny Story. It is important to note that these development sets are distinct from the test sets used in downstream benchmarks. From each validation set, we independently select the top 12% of data that had the highest impact on classification, which collectively accounted for 49.3% of the data.

[1] `https://github.com/togethercomputer/RedPajama-Data`

[2] `https://github.com/princeton-nlp/LESS`

A.2 Experimental Setup Details

A.2.1 Hyperparameters for Pre-training

All models in our experiments use the SwiGLU activation function, similar to LLaMA. We use the Adam optimizer [26] with hyperparameters set to $\beta_1 = 0.9, \beta_2 = 0.95, \varepsilon = 10^{-8}$, and weight decay fixed at 0.01. Additionally, we implement gradient norm clipping with a threshold of 1.0. A cosine learning rate schedule is employed, ensuring that the final learning rate equals 10% of the maximal learning rate (3e-4). We maintain a global batch size of 4M and vary warm-up steps based on different model sizes. To avoid the complications of insufficient training and the need for secondary adjustments, the preset steps for all pre-training processes are configured to be sufficiently long. For all training parameters see Table (9). We conduct model training based on the InternEvo framework (Team, 2023).

A.2.2 Hyperparameters for SFT

During the SFT phase, we use a cosine learning rate schedule, such that the final learning rate (1e-5) is equal to 33.3% of the maximal learning rate (3e-5). Meanwhile, no warmup is used, and the number of training steps is set to 328 (1 epoch). Other training parameters remain consistent with pre-training.

A.2.3 Computation Cost Estimation

In a series of pretraining experiments, models with varying parameter counts are evaluated for computational efficiency. For a model with 100M parameters, processing 100B tokens necessitates approximately 253 GPU hours. When the model size increased to 1B parameters, the same number of tokens required about 1388 GPU hours. Further scaling the model to 3B parameters, the token processing demands roughly 3472 GPU hours.

params	dimension	n heads	n layers	sequence length	warmup steps	maximal learning rate	preset maximal training tokens
100M	768	12	12	2048	2000	6e-4	377B
1B	2048	16	20	2048	2000	3e-4	377B
3B	3200	32	26	2048	2500	3e-4	1.1T

Table 9: Hyperparameters Setting for Pre-training Models of Different Sizes

Additionally, a 3B SFT model over 328 steps is completed within an estimated 47 GPU hours

A.2.4 Validation Sets Details

To thoroughly assess the potential impact on downstream tasks, we have meticulously chosen three unique validation datasets (pile validation sets, downstream task validation sets, and synthetic validation set), each tailored to a specific domain.

- Pile validation sets (Gao et al., 2020), including Pile-arXiv, Pile-books, Pile-OpenWebText2, and Pile-Wikipedia. These subsets are used to test the model's language modeling capabilities across a variety of knowledge-intensive tasks:

- Downstream task validation sets, which simply join prompt with a right answer from downstream benchmarks (see 4.2). These validation sets are designed to evaluate the language modeling capabilities across a variety of downstream benchmarks.

- Synthetic data validation set, including the Tiny-Story dataset (Eldan and Li, 2023). This type of validation set is primarily designed to assess a model's language modeling capabilities on synthetic texts characterized by high fluidity.

A.2.5 Downstream Tasks Details

Here, we provide a detailed description of 10 different downstream tasks in Table (10), providing insights into our model's performance in diverse linguistic contexts. We use OpenCompass (Contributors, 2023) framework to evaluate downstream tasks.

Categories	Datasets	Metric
Text Completion	StoryCloze	Acc.
Reading Comprehension	RACE-high RACE-middle	Acc.
Common-Sense QA	HellaSwag PIQA WinoGrande OpenBookQA	Acc.
Factual QA	NaturalQuestion TriviaQA	EM
Examination	MMLU	Acc.

Table 10: Downstream Benchmarks

A.3 Proxy Metric Ranking Correlation on All Validation Sets

Here we present the ranking correlations of proxy metrics on all validation sets, including 100M pre-trained model vs. 1B pre-trained model and also 1B pre-trained model vs. 3B pre-trained model.

A.3.1 Correlation Analysis of Proxy and Downstream Metrics

This study quantitatively assesses the correlation between the proxy metric (validation set PPL) of the 100M model and the downstream task metrics of the 3B SFT model. The evaluation employs a three-stage correlation analysis, using a 1B model as a bridge to handle the significant increase in training costs and improve the correlation calculation's reliability (detailed analysis see Appendix A.3.2). The ranking correlation is quantified using Pearson and Spearman Correlation coefficients, with each of them corresponding to "P" and "S" in the figures respectively. Correlation values closer to 1 indicate a higher-ranking correlation.

In the first phase, our study commences with the analysis of 14 sets of experiments, focusing on proxy metrics for models with 100M and 1B parameters, resulting in 91 paired experiments over 11 validation sets. To counter early training instability, we utilize PPL values from models trained with 100B tokens as the proxy metric. As demonstrated in Figure 5, there's a high correlation in PPL between the 100M and 1B models across most validation sets, with exceptions noted in specific datasets such as RACE-middle and TrivialQA. Generally, smaller models can predict the PPL of larger models accurately, although discrepancies in correlation coefficients are observed. Nonetheless, a clear trend is evident: an increase in PPL differences among smaller models tends to predict similar trends in larger models. Further correlation details across validation sets are presented in section A.3.

In the second phase, we conduct experiments with 7 sets of data filtering hyperparameters, each comprising proxy indicators for both 1B and 3B models. We calculate the PPL difference between each paired hyperparameter set, resulting in 21 experimental pairings on each of the seven validation sets. Considering potential early training instability, we use PPL values at the 100-billion token training mark as our metric. As illustrated in Figure 6, the PPL of the 1B and 3B models show a significant correlation across most validation datasets, with a lower correlation on RACE-middle and TrivialQA datasets, consistent with the first phase, More figures depicting the correlation on different validation sets can be seen in section A.3.

The final phase involves experiments with 7 sets of data filtering hyperparameters, each containing 3B proxy indicators and corresponding downstream evaluation metrics. As depicted in Figure 7, A Correlation value approaching -1 indicates a strong negative correlation, suggesting that lower PPL in different 3B models on the validation sets correlates with higher downstream task metrics. For most tasks, PPL can effectively predict the performance of larger models on downstream tasks. Some tasks exhibit greater variance in downstream performance, resulting in a lower correlation coefficient. Nonetheless, the graph still reveals a distinct trend: as the PPL decreases, there is a gradual improvement in the performance of downstream tasks. Detailed analysis can be seen in Appendix A.3.5.

Summarizing the previous analysis, using a 100M parameter LLM can serve as a reliable indicator for the effectiveness of pretraining corpora when applied to larger models.

A.3.2 Reason for using 1B bridge model

In an ideal scenario where computational costs are not a constraint, our target model could theoretically be as large as 7B, 10B, or even larger. However, taking into account both the computational resource limitations and the ability to manifest the model's higher-order capabilities, we have set the target model size at 3B parameters. We could directly analyze the correlation of metrics from models ranging from 100M to 3B parameters, but considering that training a single 3B model on 100B tokens requires approximately 3,472 GPU hours, which translates to a cost of about $6,944 to $17,360 based on current market GPU rates (Current market rates for A100 80GB GPUs vary between $ 2-5 / hour per gpu), the number of data points available for correlation analysis would be significantly reduced due to these computational cost constraints.

To ensure the robustness of our correlation metric analysis, we have selected a bridge model of 1B parameters that can be trained on 100B tokens at the cost of 1,388 GPU hours as a more feasible option. This allows us to increase the number of data point sets from 100M to 1B parameters to 14 sets of comparative experiments, thereby enhancing the reliability of our correlation analysis. Concurrently, the number of data point sets from 1B to 3B parameters is reduced to 7 sets of comparative experiments. However, to ensure the reliability of the metrics, we have added more checkpoint evaluations for these larger models.

We believe that under the same computational budget, conducting a greater number of experiments with varying hyperparameters on smaller models contributes more to the robustness of the correlation analysis than conducting fewer experiments on larger models.

A.3.3 100M Pre-trained vs. 1B Pre-trained

The data presented in Figure 10 show a general trend where a lower PPL in the 100M model on the validation set leads to lower PPL in the corresponding 1B model.

A.3.4 1B Pre-trained vs. 3B Pre-trained

The data presented in Figure 11 show a general trend where a lower PPL in the 1B model on the validation set leads to lower PPL in the corresponding 3B model.

A.3.5 3B Pre-trained PPL vs. 3B SFT Metric

Specifically, to address the significant variance in downstream task performance, we enhance robustness by evaluating multiple checkpoints for the same experiment, with training steps ranging from 200 to 300 billion tokens, across 25 groups. So these hyperparameters are paired to compare the PPL differences in the 3B model against the differences in downstream metrics, resulting in 300

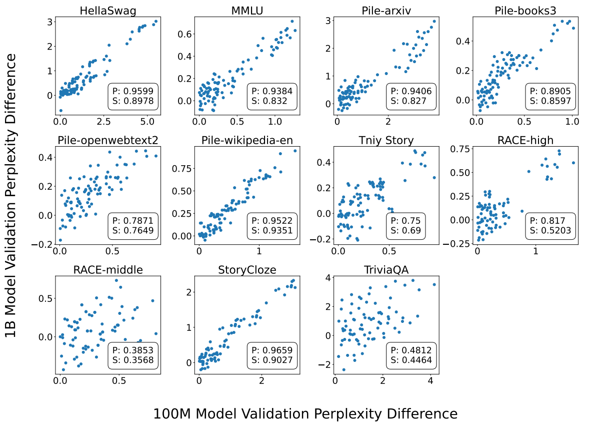


Figure 10: Validation Perplexity Difference Comparison Between 100M and 1B Model With "P" for Pearson Correlation Coefficients and "S" for Spearman Correlation Coeficients

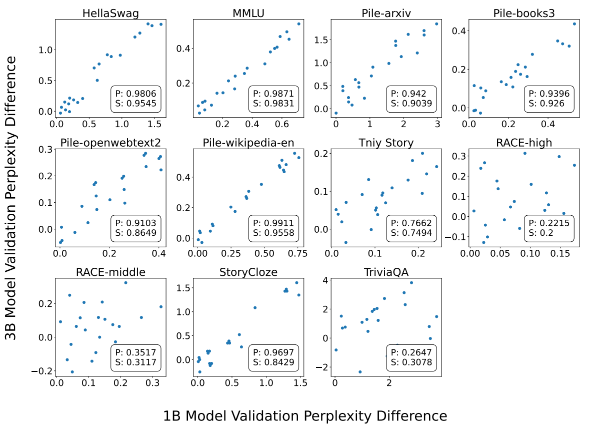


Figure 11: Validation Perplexity Difference Comparison Between 1B and 3B Model With "P" for Pearson Correlation Coefficients and "S" for Spearman Correlation Coeficients

paired experiments on each of the seven validation sets. A value approaching -1 indicates a strong negative correlation, suggesting that a smaller PPL in different 3B models on the validation set correlates with higher downstream task metrics. To further mitigate the issue of large variances, we adopt the DBSCAN method to filter out outliers, obtaining non-outlier Pearson and Spearman correlation coefficients. As depicted in Figure 7, a lower PPL in the 3B model on the validation set corresponds to superior performance on downstream tasks. For most tasks, smaller models can effectively predict the performance of larger models on downstream tasks. Some tasks exhibit greater variance in downstream performance, resulting in a lower correlation coefficient. Nonetheless, the graph still reveals a distinct trend: as the PPL decreases, the performance of downstream tasks improves gradually.

A.4 Pretraining Efficacy of Different Data Filtering Methods

In this section, we first determine the optimal thresholds and retention for different data filtering strategies based on the PPL performance of the 100M Proxy model on validation sets while also providing comparison curves for the 1B model.

Then, we will predict the performance of different filtering strategies on downstream tasks based on the PPL performance of the 100M model at the optimal thresholds.

Finally, we will pre-train the 3B model using data selected under the optimal threshold and compare downstream performances with the predictions made by the 100M model to determine the effectiveness of different data filtering strategies.

A.4.1 Loss Filtering Performace

Our analysis of the impact of data selection strategies of loss filtering at 100M and 1B parameter scale reveals varied outcomes. Strategies include no filtering and retaining the central 50% and 30% of data by loss ranking (The efficacy of the 'loss middle' data filtration strategy over 'loss bottom' or 'loss top' has been corroborated by (Marion et al., 2023), prompting us to exclusively compare the effects of two 'loss middle' thresholds against the unfiltered data).

Figure 12 presents the 100M, 1B model performance across multiple validation sets when pretraining with different tokens at various loss thresholds. We pay particular attention to the performance at the 100B token on tasks such as

MMLU (a knowledge-intensive task indicative of the model's higher-order knowledge), HellaSwag (a common-sense task, reflective of the model's common-sense reasoning), Pile-Wikipedia (a common validation set for reflecting model's breadth of knowledge) and Tiny Story (a synthetic task, representative of the model's language modeling capabilities). We summarize the partial order ranking of these validation sets in Table 1.

Synthesizing these results, we discern a notable decrease in PPL (indicating improved performance) on HellaSwag for PPL@loss middle 50%, a marked increase (indicating decreased performance) on MMLU and Wikipedia-en, and a relatively lower PPL (indicating better performance) on Tiny Story. After comprehensive consideration, we selected the loss middle 50% threshold, which corresponds to a data remaining ratio of 53.9%.

A.4.2 Wikipedia Classifier Performace

we compare the experimental effects of four sets of thresholds (0, 0.025, 0.075, 0.25). In Appendix A.1.5, we explicated the data remaining ratios under different thresholds and the threshold of 0.25 already in use for other datasets (such as RedPajama (Computer, 2023)).

Figure 13 presents the 100M & 1B model performance across multiple validation sets when pretraining with different tokens at various Wikipedia thresholds. Similar to the analysis in the previous section, we summary the partial order ranking of these validation set in Table 2.

Synthesizing these findings, we note a significant reduction in PPL (indicating performance improvement) at PPL@thresh0.075 for MMLU and Pile-Wikipedia. For HellaSwag, there is an increase in PPL (indicating worse performance, likely due to the loss of relevant data). In the case of Tiny Story, a PPL@thresh0.25 increases perplexity compared to no filtering, but PPL@thresh0.075 and PPL@thresh0.0255 initially reduce PPL, aligning with unfiltered data. This pattern underscores the nuanced effect of data filtering on text generation fluency. After comprehensive consideration, we selected a threshold of 0.075, with a data remaining ratio of 63.4%.

A.4.3 Ad Classifier Performance

Detailed ad bert classifier evaluation result is depicted in appendix A.1.3. Additionally, we explore varying ad identification thresholds (0, 0.4, 0.6, 0.8, 0.9, and 0.95) to refine our model, training across different scales: 100M, 1B, and 3B models, to optimize ad recognition capabilities.

Figure 16 presents the 100M & 1B model performance across multiple validation sets when pretraining with different tokens at various ad thresholds. Similar to the analysis in the previous section, we summarize the partial order ranking of these validation sets in Table 3.

PPL@threshold 0.95 experiences a significant increase on HellaSwag, indicating a decline in performance. Conversely, PPL@threshold 0.9 maintains a relatively lower score on MMLU, Tiny Story, and Pile-wikipedia-en, which suggests better performance. Moreover, the performance of PPL@threshold 0.9 on HellaSwag shows negligible differences when compared to other thresholds. Consequently, we have selected a threshold of 0.9, with the data retention rate being 53.9%.

A.4.4 100M Model Performance Prediction

This section is dedicated to a comparative analysis of the PPL rankings associated with the 100M model, employing various filtering strategies. The objective is to preemptively forecast the efficacy of distinct selection mechanisms when applied to downstream tasks in larger-scale models, using the smaller model as a predictive basis.

We present a ranking of the PPL scores for different data filtering strategies at their optimal thresholds for the 100M model. Additionally, we provide the PPL ranking for the 1B model for comparison. Corresponding PPL curves can be seen in Figure 17. Here we focus specifically on the results pertaining to the validation sets that are relevant to downstream tasks, as well as on the outcomes for the 'tiny story' and 'pile-wikipedia' datasets.

Table 4 is the result of 100M and 1B model with 100B tokens pretraining.

Based on the results observed, the PPL ranking of the ad filter is significantly superior to both the Wikipedia classifier and the loss filter. For the high-order knowledge understanding task MMLU, the PPL for the ad filter and Wikipedia classifier is lower than the unfiltered baseline, indicating better performance, whereas the loss filter's PPL is higher than the unfiltered baseline, indicating poorer performance. In the common sense reasoning task HellaSwag, the PPL for the ad filter is slightly higher than the unfiltered baseline, suggesting a marginal decrease in performance. Conversely, the Wikipedia classifier's PPL is significantly higher than the unfiltered baseline, indicating a substantial decrease in performance, while the loss filter's PPL is significantly lower, indicating improved performance. These results are largely consistent with the performance of the 3B model on downstream tasks as reported in Table 7. Additionally, in the following section, we will further analyze the consistency of the PPL rankings between the 100M and 1B models in conjunction with the 3B model's downstream task performance.

A.4.5 3B SFT Model Performance Evaluation

In this section, we employ the best-threshold data filtering strategies to pre-train a 3B model, followed by SFT to obtain performance metrics on downstream tasks. The outcomes are then compared with the predicted downstream task performance of the 100M model to ascertain the relative efficacy of the different data filtering methods.

Based on the results presented in Table 7, we have compiled a ranking of the effects of the various filtering strategies across several tasks in Table 5.

Compare 3B performance sorting with the previous 100M/1B PPL sorting in Table 4 we observe the following patterns:

- On the HellaSwag dataset, the performance ranking is in perfect inverse correlation with the PPL ranking of the 100M model.

- On the MMLU dataset, there is an overall inverse correlation between performance ranking and the 100M PPL ranking, with the exception of the non-filtered and loss middle 50

- On the RACE-middle and RACE-high datasets, performance rankings show overall consistency with the inverse PPL rankings of both the 100M and 1B models.

- On the TriviaQA dataset, the performance ranking is overall consistent with the inverse PPL ranking of the 100M model and perfectly consistent with the inverse PPL ranking of the 1B model.

- The StoryCloze dataset shows poorer consistency between performance ranking and the inverse PPL ranking of the 100M model, yet a overall consistency with the inverse PPL ranking of the 1B model. This may be due to the closer downstream performance across different filtering strategies for this task.

Overall, the 100M model demonstrates high consistency with the downstream performance of the larger 3B model across most tasks, and we also note high consistency between the 1B and 3B models. This supports the viability of using the 100M model to predict downstream performance for the 3B model.

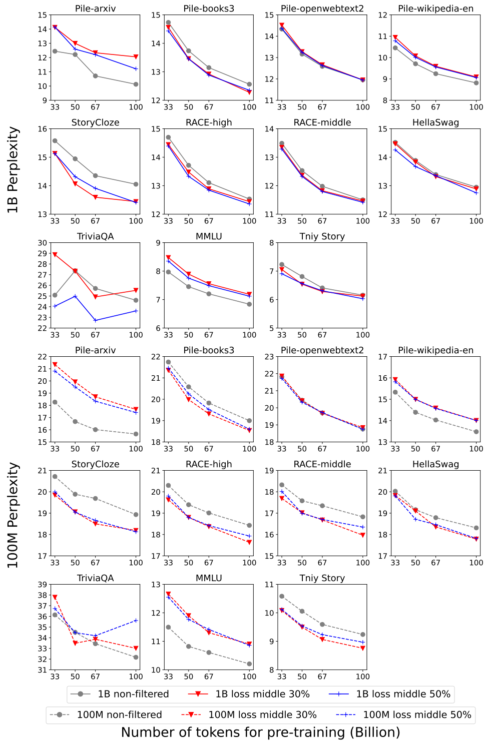


Figure 12: Validation Perplexities Comparison Between 100M & 1B Models with Moderate Loss

A.5 More Analysis

A.5.1 Comparsion to "Deep Learning Core-set Data Selection"

Our proposed Evaluation Mechanism significantly differs from the deep learning core-set data selection via proxy as described in (Coleman et al., 2019). As illustrated in Figure 9(a), the latter leverages a proxy model to generate a data selection metric, which is then used to rank and filter the data directly. The underlying assumption is that the proxy model and the target model have a high degree of consistency in the feature representation ranking of the dataset, allowing the proxy model's feature representations to substitute for those of the target model to guide data selection. However, our proposed Evaluation Mechanism employs an independent data selection model to guide the data selection process. This model may share a similar structure with the target model or be entirely heterogeneous. From this perspective, our data selection model fundamentally incorporates the concept of a proxy model as understood within the domain of supervised deep learning. However, due to the unique characteristics of unsupervised data selec-

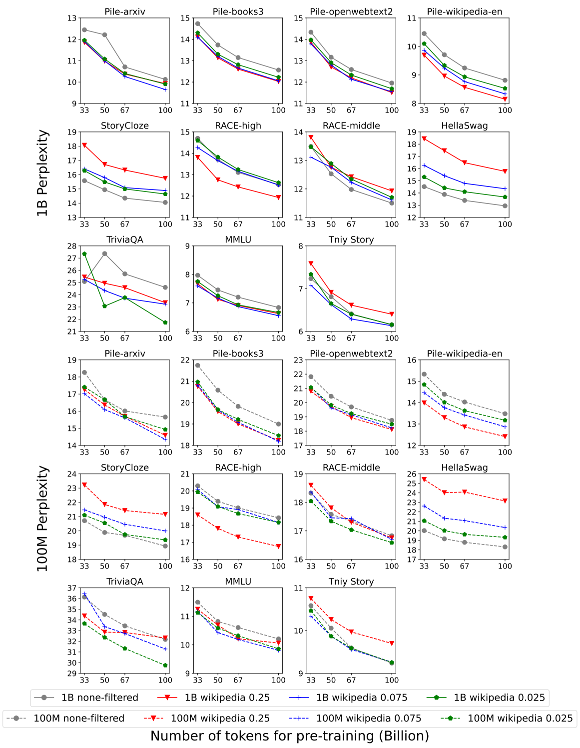


Figure 13: Validation Perplexities Comparison Between 100M & 1B Models with Wikipedia & Web

tion in LLM pretraining, the strategies employed for proxy models and Data Selection Metrics in deep learning may not be directly applicable to LLM pretraining data selection. Furthermore, we introduce a Surrogate Model and Surrogate Indicator that act as proxies for the target LLM and downstream metrics, respectively. This concept bears a resemblance to the idea of (Coleman et al., 2019), indicating a parallel in the underlying rationale.

A.5.2 Contributions of Small Proxy Model Evaluation Mechanism

1. Sufficient training to demonstrate the higher-order capabilities of small models. For instance, models ranging from 100M to 1B parameters show stable PPL at the 100B token, although there may be some instability in PPL in the early stages, A 3B model accumulates a certain amount of knowledge at the 200B token, and after SFT there is a noticeable improvement in higher-order abilities, such as those measured by MMLU. However, previous research exploring the effectiveness of data selection strategies under insufficient training conditions has obscured the manifestation of higher-order abilities, such as knowledge comprehension as measured by tasks like the MMLU.

2. Proxy indicators from PPL to post-SFT large model downstream metrics, which reveals higher-order skills like knowledge comprehension even with limited training. However prior studies using PPL from pretraining and various NLP task validation sets have shown a lack of sufficient correlation with downstream task performance, thus limiting domain-specific insights.

3. Diverse validation sets, including validation sets converted from downstream tasks, enabling early downstream performance predictions and quantifying the correlation between small model proxies and post-SFT large model downstream metrics. Total validation sets see Appendix A.2.4. However, previous research using inappropriate (in-domain) validation sets and partial downstream tasks has hindered the understanding of the impact of data selection methods on downstream tasks (refer to Section 2.2).

A.5.3 Analysis about Practical Implications and Potential Applications

This paper introduces a Small Proxy Model Evaluation Mechanism that allows the use of pre-training proxy metrics from a 100M model to predict the downstream task metrics of larger models after SFT. This rapid evaluation mechanism can significantly reduce the iteration cycles for pre-training data selection. This is meaningful for exploring the scaling laws of LLMs under higher data quality.

We provide a rough estimate of the pre-training costs involved. For a 100M model, pre-training with 100B tokens may require approximately 253 GPU hours. This means that running a set of experiments with a 100M model could cost between 506 and 1,265 dollars ((Current market rates for A100 80GB GPUs vary between $ 2-5 / hour per GPU)). When the model size increases to 3B parameters, processing these tokens would take about 3,472 GPU hours, which means that running a set of experiments with a 3B parameter model would cost between 6,944 and 17,360 dollars. By using a 100M model as a proxy for evaluation, each set of pre-training experiments could save between 6,430 and 16,095 dollars. Therefore, any team training large models that refers to our Small Proxy Model Evaluation Mechanism can save between $6,430 and $16,095 per ablation experiment group in economic costs and carbon emissions.

From the perspective of focusing on downstream task performance, this paper proposes an ad filtering strategy that generally outperforms existing

LLM pre-training data selection schemes across 10 downstream tasks. This reminds the LLM community to be aware of the potential harm of existing data selection schemes to downstream tasks. The validated ad filtering strategy can significantly shorten the cycle for the LLM community to filter high-quality data, thereby significantly reducing energy consumption. Moreover, our work is conducted on the open-source dataset RefinedWeb, and part of our work will also be made open-source in the future. In fact, utilizing our ad filtering strategy, we have trained an effective 7B parameter model that outperforms a variety of recent open-source large models.

Figure 14: The form and instructions presented to human evaluators

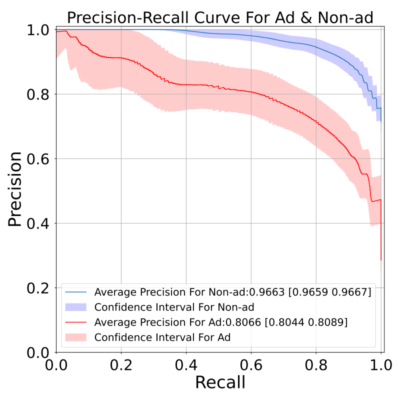


Figure 15: Effectiveness of Ad Classifier

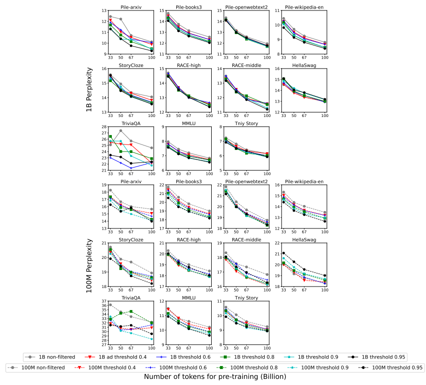


Figure 16: Validation Perplexities Comparison Between 100M & 1B Models with Ad Filtering

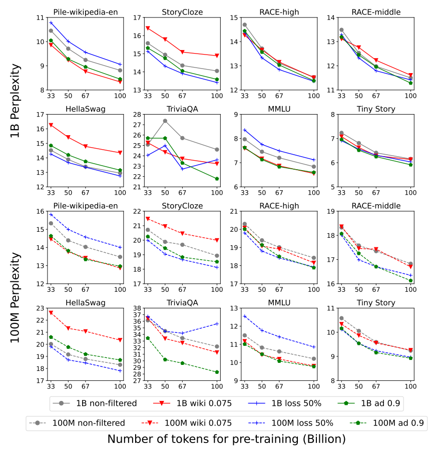


Figure 17: Validation Perplexities Comparison Between 100M & 1B Models with different Filtering Strategies

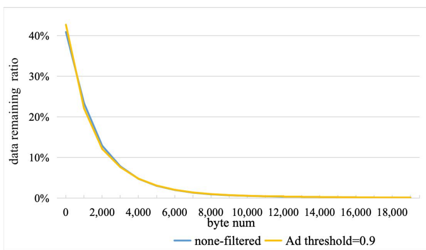


Figure 18: data length visualization before and after data filtering