Ultra-FineWeb: Efficient Data Filtering and Verification for High-Quality LLM Training Data

Anonymous ACL submission

Abstract

Data quality has become a key factor in enhancing model performance with the rapid development of large language models (LLMs). Model-driven data filtering has increasingly become a primary approach for acquiring high-quality data. However, it still faces two main challenges: (1) the lack of an efficient data verification strategy makes it difficult to provide timely feedback on data quality; and (2) the selection of seed data for training classifiers lacks clear criteria and relies heavily on human expertise, introducing a degree of subjectivity. To address the first challenge, we introduce an efficient verification strategy that enables rapid evaluation of the impact of data on LLM training with minimal computational cost. We then build upon the assumption that high-quality seed data is beneficial for LLM training, and by integrating the proposed verification strategy, we optimize the selection of positive and negative samples and propose an efficient data filtering pipeline. This pipeline not only improves filtering efficiency, classifier quality, and robustness, but also significantly reduces experimental and inference costs. By employing a lightweight *fastText*-based classifier within this pipeline, we successfully process two widely-used pre-training corpora (*FineWeb* and *Chinese FineWeb*), resulting in the creation of the higher-quality ***Ultra-FineWeb*** dataset with approximately 1.8 trillion English and 120 billion Chinese tokens. Empirical evaluations demonstrate that LLMs pre-trained on ***Ultra-FineWeb*** exhibit significant performance improvements across multiple benchmarks, validating the effectiveness of our pipeline in enhancing both data quality and training efficiency.

1 Introduction

The evolution of Large Language Models (LLMs) (Ouyang et al., 2022; Hu et al., 2024; Cai et al., 2024; Grattafiori et al., 2024; Yang et al., 2025; Team et al., 2025) has yielded transformative breakthroughs in logical reasoning, code generation, and scientific discovery (Guo et al., 2024, 2025; Lyu et al., 2025; Zhang et al., 2024). Beyond model scaling, evidence suggests that large-scale, information-intensive pre-training data is a key factor in driving the continuous improvement of LLMs' capabilities (Penedo et al., 2024; Li et al., 2024; Gunasekar et al., 2023) To construct such corpora, the prevailing paradigm involves selective filtering of massive and noisy internet data sources (Crawl, 2007). Early efforts primarily utilized heuristic-based rules (Raffel et al., 2020; Weber et al., 2025; Rae et al., 2021; Wenzek et al., 2019) and deduplication (Lee et al., 2021).

With increasing demands for data fidelity, heuristic approaches struggle to identify complex content noise, leading to suboptimal LLM performance. Consequently, model-driven filtering has emerged as a superior strategy, utilizing neural classifiers to identify and curate high-quality content (Gunasekar et al., 2023; Shao et al., 2024). This paradigm is exemplified by the success of datasets like FineWeb-edu (Penedo et al., 2024), Chinese FineWeb-edu (Yu et al., 2025), and DCLM (Li et al., 2024), integrating model-based scoring post-preprocessing to achieve both enhanced corpus purity and measurable performance gains across diverse downstream benchmarks. Despite its success, model-driven filtering faces two critical bottlenecks. First, verifying the effectiveness of a filtering strategy is computationally prohibitive, typically requiring large-scale LLM training to observe measurable gains. Second, training these classifiers requires "seed data" (initial high-quality exemplars), yet selecting such data remains an opaque process heavily reliant on subjective human expertise and heuristics.

To address these challenges, we develop a data filtering pipeline centered on an **Efficient Verification Strategy**. Instead of training LLMs from scratch, this strategy evaluates candidate corpora

by observing performance improvements during the final stages of training. Then we leverage this strategy to iteratively refine classifier seeds, guided by the premise that high-quality seed data is fundamental to LLM training. This objective refinement process further enables the deployment of a lightweight *fastText* classifier, ensuring high filtering quality and efficiency at web-scale. Utilizing this pipeline, we curate ***Ultra-FineWeb***, a superior-quality pre-training corpus comprising 1.8 trillion English and 120 billion Chinese tokens. Experimental results show that LLMs trained on ***Ultra-FineWeb*** perform excellently across multiple benchmark tasks, providing empirical validation for the effectiveness of our high-quality data filtering pipeline and its efficiency in reducing computational costs.

Our main contributions are as follows. The datasets and classifier are made publicly available.

- **Efficient Verification Strategy:** We propose a computationally efficient verification strategy that enables rapid evaluation of the impact of data on LLM training performance with minimal computational cost, significantly improving the efficiency of high-quality data filtering experiments.
- **Large-Scale High-Quality Pre-training Datasets:** We design and implement an efficient high-quality data filtering pipeline, applied to the FineWeb and Chinese FineWeb datasets, resulting in the creation of higher-quality *Ultra-FineWeb-en* and *Ultra-FineWeb-zh* datasets, collectively referred to as ***Ultra-FineWeb***. ***Ultra-FineWeb*** contains approximately 1.8 trillion English tokens and 120 billion Chinese tokens, and can facilitate high-quality LLM training.
- **Lightweight Classifier:** The *Ultra-FineWeb classifier* significantly reduces inference costs, achieving superior performance on extracted text from the same data source, thus validating the effectiveness of our proposed data filtering pipeline in enhancing data quality and training efficiency.

2 Methodology

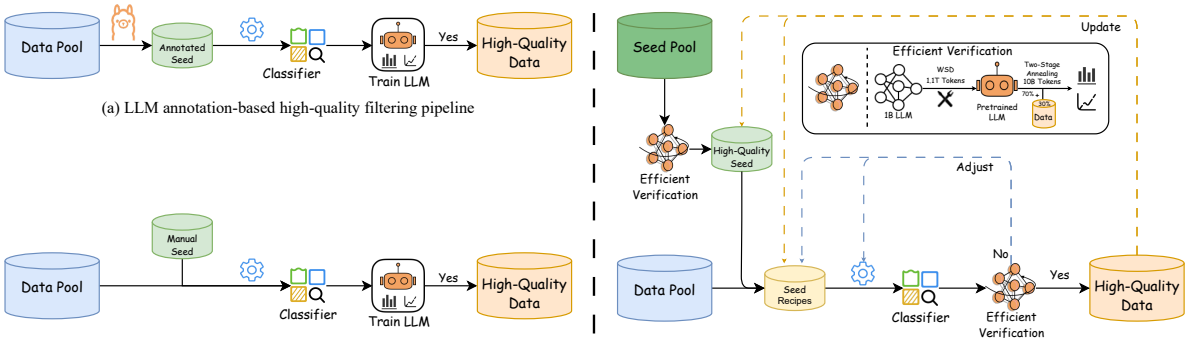
This section introduces the design and implementation of our efficient, high-quality data filtering pipeline, with the overall workflow illustrated in Figure 1(c). First, in Section 2.2, we present an Efficient Verification Strategy that significantly reduces experimental costs while ensuring the reliability of evaluation results. Subsequently, Section 2.3 outlines our methodology for selecting positive sample seed data for classifier training. Finally, Sections 2.4 and 2.5 introduce classifier training recipes and fastText-based quality filtering, respectively, which together ensure optimal data selection quality and inference efficiency.

2.1 Overall Workflow

The overall workflow of the proposed efficient verification-based high-quality filtering pipeline is illustrated in Figure 1(c). We begin by constructing an initial candidate seed pool and applying our efficient verification strategy to identify high-quality samples that significantly improve training performance. These verified samples serve as positive seeds for training a classifier, while negative samples are randomly selected from the raw data pool to create a balanced training set. During the classifier filtering stage, we sample a small subset from the raw data pool and validate the classifier's selections using our efficient verification strategy to assess its effectiveness. Based on verification results, we iteratively update the high-quality seed pool, adjust the ratio of positive and negative samples, and fine-tune classifier training hyperparameters to optimize the data selection strategy. Only classifiers demonstrating stable and reliable performance in efficient verification are deployed for full-scale data selection and subsequent model training, thereby significantly reducing computational costs while maintaining high data quality.

2.2 Efficient Verification Strategy

Validating the effectiveness of training data typically requires significant computational resources. For instance, training a 1 billion (B) LLM on 100B tokens requires approximately 1,200 H100 GPU hours (equivalent to 64 GPUs running continuously for nearly 19 hours). This computational burden becomes particularly prohibitive when iteratively developing high-quality data classifiers. Moreover, large-scale training validation proves impractical for smaller datasets, as models trained with limited token counts fail to exhibit statistically significant performance differences, with training instability further compromising result reliability. This limitation is evident in our comparative analysis of FineWeb and FineWeb-edu (Penedo et al., 2024). When trained from scratch with 8 billion to-



(a) LLM annotation-based high-quality filtering pipeline

(b) Manual seed-based high-quality filtering pipeline

(c) Efficient verification-based high-quality filtering pipeline

Figure 1: Comparison of High-Quality Data Filtering Pipelines. Traditional model-based data filtering methods (a) and (b) rely on human expertise for seed data selection and lack data quality verification.

kens, FineWeb-edu achieves superior performance on HellaSwag (Zellers et al., 2019), while at 380 billion tokens, FineWeb demonstrates better results across multiple benchmarks, including Winogrande (Sakaguchi et al., 2021), HellaSwag (Zellers et al., 2019), and PIQA (Bisk et al., 2020), highlighting the inconsistency in evaluation outcomes based on training scale[1].

Inspired by Llama 3.1 (Dubey et al., 2024), we design an Efficient Verification Strategy. We begin by training a 1B LLM on 1.1 trillion (T) tokens using a WSD scheduler (Hu et al., 2024) (comprising stable training on 1T tokens, followed by decay training on 0.1T tokens). Based on this pretrained LLM, we then implement a two-stage annealing process with 10B tokens, allocating 30% of the weight to the verification data, while keeping the remaining 70% for the default mixed data ratio. Model details and training hyperparameters can be found in Appendix A. This optimized strategy reduces computational costs from 1,200 to approximately 110 H100 GPU hours (see Table 4), significantly improving the efficiency and iterability of the filtering process. This strategy allows for efficient assessment of the impact of verification data across various evaluation dimensions. To validate the reliability of this strategy, we compare the results of training 100B tokens from scratch on the 1B LLM using FineWeb and FineWeb-edu, respectively. As shown in Table 6, the results follow similar trends, with further experimental analysis provided in Appendix C.

[1] https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

2.3 Classifier Training Seeds

The effectiveness of high-quality data classifiers fundamentally depends on the selection of superior positive training samples. As illustrated in Figure 1(a), datasets such as FineWeb-edu (Penedo et al., 2024), Chinese-FineWeb-edu (Yu et al., 2025), and CCI3-HQ (Wang et al., 2024) employ LLM annotation-based frameworks to partially label source-consistent data, generating "seed data". In contrast, Figure 1(b) demonstrates manual seed-based filtering (DCLM's (Li et al., 2024)) pipeline, which relies on manual curation for positive sample selection, focusing specifically on instruction-formatted data by incorporating samples from OpenHermes 2.5 (OH-2.5) (Teknium, 2023) and high-quality posts from the `r/ExplainLikeImFive` (ELI5) subreddit.

Although both pipelines demonstrate distinct advantages in selecting positive samples, they are accompanied by inherent limitations. The LLM annotation-based pipeline can effectively filter high-quality samples from source-consistent data, but its performance is constrained by the scoring criteria of the LLM, potentially introducing systematic biases and annotation noise. Furthermore, classifiers trained exclusively on source-consistent data often exhibit limited generalization capabilities and poor robustness. Conversely, manual curation faces significant methodological challenges: the effectiveness of seed data is difficult to assess before classifier training, and its validation relies heavily on the performance of LLMs trained on the filtered data. These constraints lead to high computational costs and reduced adaptability across different tasks.

Based on these considerations, we propose a key assumption: high-quality seed data that enhances

LLM performance will yield classifiers capable of identifying similarly beneficial training data. As illustrated in Figure 1 (c), we implement our Efficient Verification Strategy to rapidly evaluate and validate seed data quality within the candidate pool, ensuring the selection of samples that can improve LLM training results. This pipeline not only ensures superior data quality but also optimizes filtration efficiency, thereby generating more reliable positive samples for classifier training. Furthermore, to enhance classifier robustness, we expand negative sample selection beyond source-consistent data. Experimental results further demonstrate that incorporating diverse data sources for negative samples can improve the generalizability of the classifier.

2.4 Classifier Training Recipes

We evaluate a large pool of candidate seed data and ultimately select those with clear effectiveness as positive samples. The positive samples include: (1) LLM-annotated data with scores above 4[2,3]; (2) instruction-formatted datasets such as OH-2.5 and ELI5; (3) authentic textbook data; (4) LLM-synthesized educational data; and (5) high-quality web content obtained through targeted crawling. For negative samples, we incorporate raw data from diverse sources, including English corpora (FineWeb (Penedo et al., 2024), C4 (Raffel et al., 2020), Dolma (Soldaini et al., 2024), Pile (Gao et al., 2020), and RedPajama (Weber et al., 2025)) and Chinese datasets (IndustryCorpus2 (Shi et al., 2024), MiChao (Liu et al., 2023), WuDao (BAAI, 2023), SkyPile (Wei et al., 2023), WanJuan (Qiu et al., 2024), ChineseWebText (Chen et al., 2023), TeleChat (He et al., 2024), and CCI3 (Wang et al., 2024)) in the initial iteration. To maintain dataset diversity and balance, we implement a uniform distribution strategy, with underrepresented categories undergoing 3-5 rounds of strategic resampling.

Subsequently, we conduct a single iteration of the classifier, utilizing its current predictions as training data for the next round. However, empirical results indicate that the iterative process only contributed meaningfully in the first round, as subsequent updates do not yield further performance improvements and, in some cases, even lead to a decrease in LLM performance. Our analysis shows that classifier improvement primarily depends on the seed data selection, rather than iterative refinement using inferred samples. Interestingly, we find that intersecting high-quality data filtered by multiple classifiers consistently improves LLM performance.

2.5 FastText-based Quality Filtering

Current high-quality data classifiers are primarily divided into LLM-based (Penedo et al., 2024; Yu et al., 2025; Wang et al., 2024) and fastText-based (Li et al., 2024; Shao et al., 2024; Guo et al., 2024) methods. While LLM-based classifiers are effective, they need significantly higher inference costs. To address this, we adopt a fastText-based classifier, which significantly reduces inference costs while maintaining competitive performance under certain conditions. This approach not only minimizes resource consumption but also speeds up data filtering experiments. For instance, processing 15T tokens with an LLM-based classifier requires approximately 6,000 H100 GPU hours, whereas fastText completes the same task in 1,000 hours on a CPU-only machine (80 CPUs), without any GPU, greatly improving efficiency. Notably, most of our large-scale experiments are conducted in a distributed manner using a Spark[4] cluster.

For data preprocessing, we implement several key steps, including removing redundant empty lines and extra spaces, stripping diacritics, and converting all English text to lowercase. Additionally, we adopt the DeepSeek-V2 tokenizer (Liu et al., 2024), which outperforms traditional tokenization methods (such as space-based tokenization for English and Jieba[5] for Chinese). Meanwhile, we preserve structural information such as `\n`, `\t`, and `\r`. To ensure dataset integrity and balance, the final training set comprised 600K samples, evenly split between positive and negative examples.

For training details, we trained a fastText classifier with a vector dimension of 256, a learning rate of 0.1, a maximum word n-gram length of 3, a minimum word occurrence threshold of 5, and a total of 3 training epochs. Additionally, during inference, we maintain the default threshold of 0.5 to simplify operations and ensure experimental consistency, avoiding the need for additional tuning steps.

[2] https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu-llama3-annotations

[3] https://huggingface.co/datasets/BAAI/CCI3-HQ-Annotation-Benchmark

[4] https://spark.apache.org/

[5] https://pypi.org/project/jieba/

3 Experiments

In this section, we first detail the experimental settings in Section 3.1, including the training configuration, data composition, and evaluation metrics. Then, in Section 3.2, we present the experimental results, highlighting the performance comparisons between individual datasets and mixed datasets, and analyze the performance of the proposed method. These results demonstrate that *Ultra-FineWeb*, obtained through our efficient data filtering pipeline, exhibits superior quality to other datasets derived from the same data source, with the corresponding trained models achieving enhanced performance.

3.1 Experimental Setting

In our experiments, all models are trained using the open-source Megatron-LM library (Shoeybi et al., 2019). We utilize the MiniCPM-1.2B model architecture with the MiniCPM3-4B tokenizer. Each experiment involves training on approximately 100B tokens. We employ the Lighteval (Fourrier et al., 2023) library for model evaluation, mirroring the setup used with FineWeb (Penedo et al., 2024) and CCI3-HQ (Wang et al., 2024). All evaluation metrics are based on a zero-shot setting. The evaluation metrics include English and Chinese metrics. Detailed configurations, dataset composition, and evaluation metrics are provided in Appendix B.

3.2 Results and Analysis

Individual Dataset Results. We compare the performance of models trained on 100B tokens using data extracted from the FineWeb and Chinese FineWeb sources, using three different approaches: raw data, LLM-based classifiers (-edu), and the fastText-based classifier trained via the Efficient Data Filtering Pipeline (Ultra-). As shown in Tables 1, on the English Metrics, Ultra-FineWeb-en demonstrates significant improvements in performance on multiple tasks, including MMLU, ARC-C, ARC-E, CommonSenseQA, and OpenBookQA. Specifically, Ultra-FineWeb outperforms FineWeb in these tasks, with only a slight drop of 0.15 percentage points (*pp*) in HellaSwag compared to FineWeb, but a 0.6*pp* improvement over FineWeb-edu. The English average score for Ultra-FineWeb-en (45.891*pp*) is 3.61*pp* higher than that of FineWeb (42.287*pp*) and 1.3*pp* higher than FineWeb-edu (44.560*pp*). On the Chinese metrics, Ultra-FineWeb-zh also outperforms both FineWeb-zh and FineWeb-edu-zh on C-Eval and CMMLU. Specifically, Ultra-FineWeb-zh improves by 0.31*pp* and 3.65*pp* over Chinese FineWeb and Chinese FineWeb-edu-v2 on C-Eval and CMMLU, respectively, and by 0.09*pp* and 0.13*pp* compared to FineWeb-edu-zh. The Chinese average score for Ultra-FineWeb-zh increases by 1.98*pp* and 0.61*pp*, respectively, compared to FineWeb-zh and FineWeb-edu-zh. These results indicate that our proposed High-Quality Data Filtering Pipeline significantly improves data quality, leading to notable improvements in model performance. Additionally, we evaluate the performance at each training checkpoint. As shown in Figure 2, Ultra-FineWeb-en surpasses both FineWeb and FineWeb-edu early in the training process, while Ultra-FineWeb-zh demonstrates a marked improvement in Chinese average scores after 40B tokens of training.

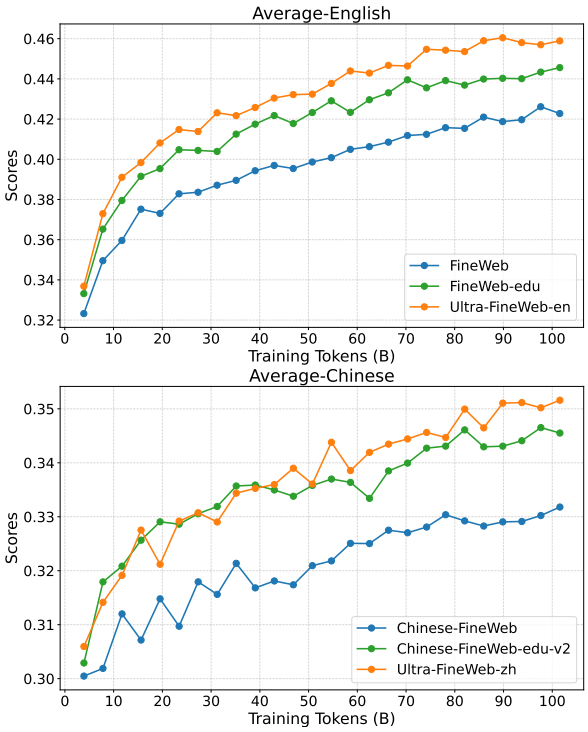


Figure 2: Average scores at each checkpoint for different individual datasets.

Mixed Dataset Results. In the mixed data experiments, we compare the model performance on different evaluation sets after training 100B tokens with the original data, LLM-based classifier-extracted edu data, and our Ultra-FineWeb dataset, using the same training configuration, and combining 60% English, 30% Chinese, and 10% code. As shown in Table 2, Ultra-FineWeb demonstrates significant performance improve-

Metrics	FineWeb	FineWeb-edu	Ultra-FineWeb-en
MMLU	28.84	31.80$_{+2.96}$	**32.24**$_{+3.4}$
ARC-C	25.17	34.56$_{+9.39}$	**35.67**$_{+10.5}$
ARC-E	59.18	69.95$_{+10.77}$	**70.62**$_{+11.44}$
CommonSenseQA	34.32	31.53$_{-2.79}$	**36.45**$_{+2.13}$
HellaSwag	**42.91**	42.17$_{-0.74}$	42.76$_{-0.15}$
OpenbookQA	22.20	25.20$_{+3.00}$	**26.20**$_{+4.00}$
PIQA	73.29	72.14$_{-1.15}$	**73.67**$_{+0.38}$
SIQA	38.95	38.13$_{-0.82}$	**39.61**$_{+0.66}$
Winogrande	55.64	55.56$_{-0.08}$	**55.80**$_{+0.16}$
Average$_{English}$	42.278	44.560$_{+2.282}$	**45.891**$_{+3.613}$
Metrics	Chinese-FineWeb	Chinese-FineWeb-edu-v2	Ultra-FineWeb-zh
C-Eval	33.95	34.17$_{+0.22}$	**34.26**$_{+0.31}$
CMMLU	32.41	34.93$_{+2.52}$	**36.06**$_{+3.65}$
Average$_{Chinese}$	33.18	34.55$_{+1.370}$	**35.16**$_{+1.980}$

Table 1: Comparison of individual results on English and Chinese datasets.

ments on multiple benchmarks. The average English score is 2.905pp higher than FineWeb$_{mix}$ (41.366pp) and 0.538pp higher than FineWeb-edu$_{mix}$ (43.733pp). For the Chinese evaluation set, Ultra-FineWeb achieves a 1.715pp advantage over than FineWeb$_{mix}$ (32.01pp), while showing a marginal 0.025pp decrease compared to FineWeb-edu$_{mix}$ (33.75pp). This minor discrepancy may stem from dataset weight setting or inherent training instability, warranting further investigation in future studies. The comprehensive analysis reveals Ultra-FineWeb's superior performance over both baseline and LLM-filtered datasets, demonstrating significant overall score improvements. Despite task-specific fluctuations, Ultra-FineWeb, generated through our Efficient Data Filtering Pipeline, consistently delivers effective performance enhancements. The line charts of checkpoint evaluations are shown in Figure 3. In the early training phases, Ultra-FineWeb and FineWeb-edu$_{mix}$ exhibit comparable performance, but both outperform FineWeb$_{mix}$. Notably, Ultra-FineWeb starts to surpass FineWeb-edu$_{mix}$ after training approximately 60B tokens. As for Chinese evaluation metrics, both Ultra-FineWeb and FineWeb-edu$_{mix}$ demonstrate training fluctuations while maintaining substantial advantages over FineWeb$_{mix}$ throughout the training process.

Multi-Turn Training Recipes Results. To verify the impact of multiple iterations on classifier performance, we implement three rounds of iterations for both English and Chinese classifiers. The initial iteration utilizes the selected high-quality seed data for positive samples and multi-source original data for negative samples. The second iteration involves using the classifier from the first round to process the negative samples, and the inferred positive and negative samples are incorporated into the next round of training data. The third iteration involves updating the classifier with more precisely identified samples from the second round. Experimental results (Tables 3) indicate that second-iteration classifiers achieved superior performance across multiple tasks compared to the first-iteration. Notably, English classifiers demonstrate significant improvements in MMLU, ARC-C, and OpenbookQA tasks, with an average score increase of 3.613 percentage points (pp) over both the first iteration and original FineWeb dataset, reaching 45.89pp. However, the third iteration, which focused solely on updating samples from original source data, failed to yield additional performance gains. In fact, there were slight declines in some tasks, such as HellaSwag and PIQA. For the Chinese data, the second iteration of Ultra-FineWeb-zh also shows notable improvements in CMMLU and C-Eval. However, similar to the English results, the third iteration provided only marginal overall improvements, with no significant gains in specific tasks. This suggests that iterative sample refinement through enhanced classifiers alone is insufficient for achieving further performance improvements.

To further substantiate the robustness and effectiveness of our approach, we provide extended analyses in the Appendix. Specifically, Appendix C offers a rigorous validation of the *Efficient Verification Strategy*, including its stability, sensitivity to verification ratios, and model-agnostic behavior across different training cutoffs. Furthermore,

Metrics	FineWeb$_{mix}$	FineWeb-edu$_{mix}$	Ultra-FineWeb
MMLU	28.50	**30.95**$_{+2.45}$	30.94$_{+2.44}$
ARC-C	24.15	32.34$_{+8.19}$	**33.36**$_{+9.21}$
ARC-E	55.60	67.13$_{+11.53}$	**67.97**$_{+12.37}$
CommonSenseQA	36.20	35.79$_{-0.41}$	**37.18**$_{+0.98}$
HellaSwag	**40.28**	40.21$_{-0.07}$	39.65$_{-0.63}$
OpenbookQA	21.60	23.80$_{+2.20}$	**24.40**$_{+2.80}$
PIQA	71.11	**71.22**$_{+0.11}$	70.08$_{-1.03}$
SIQA	39.76	39.20$_{-0.56}$	**40.48**$_{+0.72}$
Winogrande	**55.09**	52.96$_{-2.13}$	54.38$_{-0.71}$
C-Eval	33.79	**34.32**$_{+0.53}$	34.10$_{+0.31}$
CMMLU	30.23	33.18$_{+2.95}$	**33.35**$_{+3.12}$
Average$_{English}$	41.366	43.733$_{+2.367}$	**44.271**$_{+2.905}$
Average$_{Chinese}$	32.010	**33.750**$_{+1.740}$	33.725$_{+1.715}$
Average	39.665	41.918$_{+2.253}$	**42.354**$_{+2.689}$

Table 2: Comparison of results on mixed datasets.

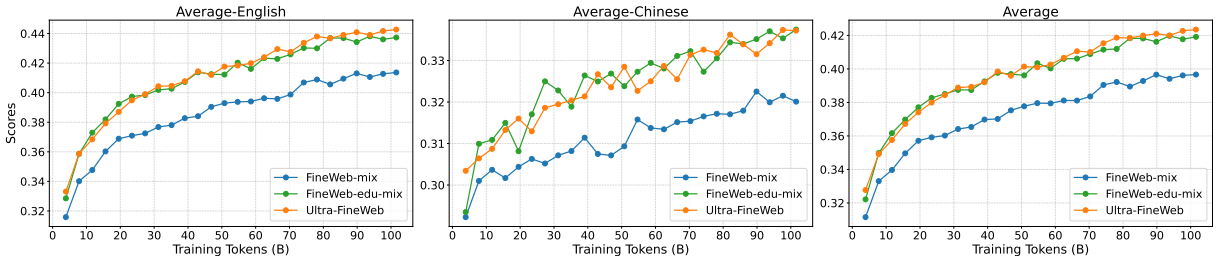


Figure 3: Average scores at each checkpoint for different mixed datasets.

Appendix D presents detailed ablation studies on multi-classifier intersections, token length distributions, and performance estimations based on scaling laws.

4 Related Work

The success of LLMs largely depends on the availability of large-scale, high-quality pretraining corpora, which provide models with rich knowledge and reasoning capabilities. Common Crawl (Crawl, 2007) has served as the foundation data source for LLM development, and to meet the growing data requirements for training larger models, a vast amount of pretraining corpora have been made open source. Early efforts, such as C4 (Raffel et al., 2020) with 160B tokens and Pile (Gao et al., 2020) with 300B tokens, provide critical resources for early model pretraining. In recent years, substantially larger corpora have emerged, including RefinedWeb (Penedo et al., 2023) with 600B tokens, Dolma (Soldaini et al., 2024) with 3T tokens, FineWeb (Penedo et al., 2024) with 15T tokens, RedPajama-v2 (Weber et al., 2025) with 30T tokens, and DCLM (Li et al., 2024) with 240T tokens, significantly advancing LLM development, fostering community collaboration, and establishing new benchmarks for innovation. Meanwhile, Chinese pretraining corpora have also been rapidly developed, such as ChineseWebText (Chen et al., 2023) with 50B tokens, WuDao (BAAI, 2023) with 120B tokens, IndustryCorpus2 (Shi et al., 2024) with 200B tokens, and CCI3 (Wang et al., 2024) with 200B tokens. However, despite progress in traditional data processing methods (such as heuristic filtering and deduplication) during the early stages, the processed data still often contains noise and unstructured content. With the continuous scaling up of models and increasing demands for data quality, these methods have become insufficient to meet current requirements.

To address these challenges, model-driven data filtering strategies have gradually become an effective approach to improving data quality in recent years. These approaches are primarily implemented during the final stages of large-scale data preprocessing, aiming to filter high-quality and high-value samples from massive datasets to further enhance model performance. Traditional quality filtering techniques (Penedo et al., 2024; Wang et al., 2024; Li et al., 2024; Yu et al., 2025) typically train classifiers to distinguish between high-quality data (such as textbook text) and low-quality data (such

Metrics	FineWeb	fastText-en-v1	Ultra-FineWeb-en	fastText-en-v3
MMLU	28.84	32.30$_{+3.46}$	32.24$_{+3.40}$	**32.29**$_{+3.45}$
ARC-C	25.17	**35.67**$_{+10.50}$	**35.67**$_{+10.50}$	35.07$_{+9.9}$
ARC-E	59.18	70.33$_{+11.15}$	**70.62**$_{+11.44}$	70.54$_{+11.36}$
CommonSenseQA	34.32	32.27$_{-2.05}$	36.45$_{+2.13}$	**36.55**$_{+2.23}$
HellaSwag	**42.91**	42.82$_{-0.09}$	42.76$_{-0.15}$	42.62$_{-0.29}$
OpenbookQA	22.20	24.40$_{+2.20}$	**26.20**$_{+4.00}$	**26.20**$_{+4.00}$
PIQA	73.29	72.09$_{-1.20}$	**73.67**$_{+0.38}$	72.53$_{-0.76}$
SIQA	38.95	38.59$_{-0.36}$	**39.61**$_{+0.66}$	39.41$_{+0.46}$
Winogrande	55.64	55.09$_{-0.55}$	55.80$_{+0.16}$	**55.92**$_{+0.28}$
$Average_{English}$	42.278	44.840$_{+2.562}$	**45.891**$_{+3.613}$	45.681$_{+3.403}$
Metrics	Chinese-FineWeb	fastText-zh-v1	Ultra-FineWeb-zh	fastText-zh-v3
C-Eval	33.95	33.63$_{-0.32}$	**34.26**$_{+0.31}$	**34.26**$_{+0.31}$
CMMLU	32.41	35.82$_{+3.41}$	**36.06**$_{+3.65}$	35.07$_{+2.66}$
$Average_{Chinese}$	34.035	35.390$_{+1.355}$	**35.875**$_{+1.840}$	34.26$_{+0.225}$

Table 3: Comparison of results on English and Chinese datasets with multiple iterations.

as raw web text), subsequently filtering out samples with lower inference scores. Additionally, data filtering methods based on perplexity (Muennighoff et al., 2024; Wenzek et al., 2019), and strategies using pre-trained LLMs to evaluate multiple dimensions of data quality through prompts (Sachdeva et al., 2024; Wettig et al., 2024), have been introduced. These advancements have greatly expanded the range of data filtering methods available.

The common trend of these methods is to obtain higher-quality data by reducing computational costs. By optimizing the filtering process and reducing inference resource consumption, not only is dataset quality improved, but data processing efficiency is also accelerated. This optimization enables LLMs to access superior training corpora, facilitating enhanced model performance with reduced training token requirements.

5 Conclusion

In this paper, we construct a higher-quality ***Ultra-FineWeb*** dataset (including English data *Ultra-FineWeb-en*, approximately 1.8T tokens, and Chinese data *Ultra-FineWeb-zh*, approximately 120B tokens, totaling approximately 1.9T tokens). This dataset is based on the FineWeb and Chinese FineWeb datasets, utilizing our proposed efficient data filtering pipeline. Through rigorous experimental evaluations, we demonstrate that Ultra-FineWeb-en and Ultra-FineWeb-zh outperform FineWeb-edu and Chinese FineWeb-edu-v2 when used for small-scale model training from scratch. Additionally, we show the effectiveness of the high-quality data filtered by our classifier on the DCLM-Pool and MAP-CC datasets, further confirming the reliability and effectiveness of our proposed pipeline. These results indicate that classifiers based on our efficient data filtering pipeline can select higher-quality data with reduced computational cost, thereby improving model training performance. We provide a detailed description of the implementation of our efficient data filtering pipeline, especially the efficient verification strategy driven by classifiers in the pipeline. This strategy enables reliable assessment of training data impact on LLM performance while maintaining minimal computational requirements. Furthermore, we present detailed methodologies for classifier seed data selection, training recipes, and FastText model training configuration, ensuring experimental reproducibility and result transparency. This study aims to provide novel insights and methodologies for high-quality data filtering, offering valuable references for data quality optimization in future LLM training processes, and contributing to the further development of LLMs.

Limitations

While our work demonstrates the effectiveness of an efficient and scalable data filtering and verification pipeline, several limitations remain. First, our experiments adopt a fixed classification threshold ($thr = 0.5$), which provides a stable and practical operating point in large-scale settings but may not be optimal across all data sources or iterative filtering stages. Second, our evaluation primarily focuses on general-domain web data, and the behavior of the proposed pipeline in highly spe-

cialized domains (e.g., mathematics, code, or legal text) has not been systematically explored. Finally, data quality is mainly assessed through downstream model performance, which, while practical, remains dependent on specific model architectures and training configurations. We view these limitations as natural boundaries of the current study, and addressing them may further extend the applicability of efficient verification for large-scale language model pretraining.

Ethical Considerations and Broader Impacts

UltraFineWeb is developed using publicly available datasets, with a focus on data quality improvement for natural language processing tasks. Data collection and preprocessing adhere to guidelines that minimize potential harm, ensuring no personally identifiable information (PII) is included and mitigating biased or inappropriate content. UltraFineWeb aims to advance language model capabilities across multilingual and cross-lingual contexts, benefiting research, education, and industry. The dataset enhances data quality and diversity to support fairer, more inclusive AI systems. However, potential risks include misuse of generated content, intellectual property concerns, and the propagation of biases. To address these risks, we implement data curation protocols, provide usage guidelines, and engage with the research community to promote transparency and accountability.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

BAAI. 2023. Wudao corpus.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, and 1 others. 2024. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*.

Jianghao Chen, Pu Jian, Tengxiao Xi, Dongyi Yi, Qianlong Du, Chenglin Ding, Guibo Zhu, Chengqing Zong, Jinqiao Wang, and Jiajun Zhang. 2023. ChineseWebText: Large-scale high-quality chinese web text extracted with effective evaluation model. *arXiv preprint arXiv:2311.01149*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

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*.

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

Common Crawl. 2007. Common crawl. https://commoncrawl.org.

Xinrun Du, Zhouliang Yu, Songyang Gao, Ding Pan, Yuyang Cheng, Ziyang Ma, Ruibin Yuan, Xingwei Qu, Jiaheng Liu, Tianyu Zheng, and 1 others. 2024. Chinese tiny llm: Pretraining a chinese-centric large language model. *arXiv preprint arXiv:2404.04167*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Clémentine Fourrier, Nathan Habib, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. 2023. Lighteval: A lightweight framework for llm evaluation.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, and 1 others. 2020. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, and 1 others. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Yu Wu, YK Li, and 1 others. 2024. Deepseek-coder: When the large language model meets programming–the rise of code intelligence. *arXiv preprint arXiv:2401.14196*.

Zhongjiang He, Zihan Wang, Xinzhang Liu, Shixuan Liu, Yitong Yao, Yuyao Huang, Xuelong Li, Yongxiang Li, Zhonghao Che, Zhaoxi Zhang, and 1 others. 2024. Telechat technical report. *arXiv preprint arXiv:2401.03804*.

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*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, and 1 others. 2024. MiniCPM: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2021. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023. Cmmlu: Measuring massive multitask language understanding in chinese. *Preprint*, arXiv:2306.09212.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, and 1 others. 2024. Datacomp-lm: In search of the next generation of training sets for language models. *arXiv preprint arXiv:2406.11794*.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, and 1 others. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*.

Yidong Liu, FuKai Shang, Fang Wang, Rui Xu, Jun Wang, Wei Li, Yao Li, and Conghui He. 2023. Michao-huafen 1.0: A specialized pre-trained corpus dataset for domain-specific large models. *arXiv preprint arXiv:2309.13079*.

Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, and 1 others. 2024. Starcoder 2 and the stack v2: The next generation. *arXiv preprint arXiv:2402.19173*.

Chengqi Lyu, Songyang Gao, Yuzhe Gu, Wenwei Zhang, Jianfei Gao, Kuikun Liu, Ziyi Wang, Shuaibin Li, Qian Zhao, Haian Huang, and 1 others. 2025. Exploring the limit of outcome reward for learning mathematical reasoning. *arXiv preprint arXiv:2502.06781*.

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*.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. 2024. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, and 1 others. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*.

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*.

Jiantao Qiu, Haijun Lv, Zhenjiang Jin, Rui Wang, Wenchang Ning, Jia Yu, ChaoBin Zhang, Zhenxiang Li, Pei Chu, Yuan Qu, and 1 others. 2024. Wanjuan-cc: A safe and high-quality open-sourced english webtext dataset. *arXiv preprint arXiv:2402.19282*.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, and 1 others. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

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

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

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.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *Preprint*, arXiv:1904.09728.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Xiaofeng Shi, Lulu Zhao, Hua Zhou, and Donglin Hao. 2024. Industrycorpus2.

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*.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, and 1 others. 2024. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, and 1 others. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

MiniCPM Team, Chaojun Xiao, Yuxuan Li, Xu Han, Yuzhuo Bai, Jie Cai, Haotian Chen, Wentong Chen, Xin Cong, Ganqu Cui, and 1 others. 2025. Minicpm4: Ultra-efficient llms on end devices. *arXiv preprint arXiv:2506.07900*.

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

Liangdong Wang, Bo-Wen Zhang, Chengwei Wu, Hanyu Zhao, Xiaofeng Shi, Shuhao Gu, Jijie Li, Quanyue Ma, TengFei Pan, and Guang Liu. 2024. Cci3. 0-hq: a large-scale chinese dataset of high quality designed for pre-training large language models. *arXiv preprint arXiv:2410.18505*.

Maurice Weber, Dan Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, and 1 others. 2025. Redpajama: an open dataset for training large language models. *Advances in Neural Information Processing Systems*, 37:116462–116492.

Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, Haihua Yang, Biye Li, Cheng Cheng, Weiwei Lü, Rui Hu, and 1 others. 2023. Skywork: A more open bilingual foundation model. *arXiv preprint arXiv:2310.19341*.

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*.

Chaojun Xiao, Jie Cai, Weilin Zhao, Guoyang Zeng, Xu Han, Zhiyuan Liu, and Maosong Sun. 2024. Densing law of llms. *arXiv preprint arXiv:2412.04315*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report.

Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. 2022. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv preprint arXiv:2203.03466*.

Yijiong Yu, Ziyun Dai, Zekun Wang, Wei Wang, Ran Chen, and Ji Pei. 2025. Opencsg chinese corpus: A series of high-quality chinese datasets for llm training. *arXiv preprint arXiv:2501.08197*.

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*.

Kaiyan Zhang, Sihang Zeng, Ermo Hua, Ning Ding, Zhang-Ren Chen, Zhiyuan Ma, Haoxin Li, Ganqu Cui, Biqing Qi, Xuekai Zhu, and 1 others. 2024. Ultramedical: Building specialized generalists in biomedicine. *arXiv preprint arXiv:2406.03949*.

Fan Zhou, Zengzhi Wang, Qian Liu, Junlong Li, and Pengfei Liu. 2024. Programming every example: Lifting pre-training data quality like experts at scale. *arXiv preprint arXiv:2409.17115*.

A Implementation Details for Efficient Verification

All Efficient Verification experiments are trained using the open-source Megatron-LM library. We utilize the MiniCPM-1.2B model architecture with the MiniCPM-3-4B tokenizer. The model training utilizes the MiniCPM-3-4B training corpus, and the WSD scheduler. We train the model from scratch with 1.1T tokens (1T for the stable stage and 0.1T for the decaying stage). Based on this pretrained model, we further perform a two-stage annealing training with 10B tokens, allocating 30% of the weight to the verification data, while keeping the remaining 70% for the default mixed data ratio. Key training parameters include a sequence length of 4096, weight decay of 0.1, and a gradient clipping threshold of 1.0. We employed a global batch size of 512. For larger datasets, we train for a total of 5000 steps (approximately 10B tokens). For smaller datasets, we compute the total training steps based on the actual token size of the data and typically allowed the validation data to undergo 3-5 training epochs (n_{epoch}).

The training steps calculation formula is as follows:

$$Total\ Iter = \max\left(\frac{Total\ Token}{Global\ BS \times Seq\ Len}, 5000\right)$$

Where $Total\ Token$ is calculated as:

$$Total\ Token = \frac{Curr\ Data\ Token \times n_{epoch}}{0.3}$$

To reduce the cost of baseline experiments, we typically choose training steps of 100, 500, 1,000, 2,500, or 5,000, balancing experimental accuracy and computational resource consumption. This means that for datasets of different scales, we dynamically adjust the training steps based on the data tokens required for training. It is important to note that the training steps for the baseline experiments are also dynamically adjusted based on the corresponding dataset size. Additionally, we set the warmup fraction to 0.1, and the annealing phase used an exponential decay approach, with the maximum learning rate to 1e-3 and the minimum learning rate to 5e-5. To enhance training stability, we use Maximal Update Parameterization (MuP).

Strategy	GPU Hours
100B from scratch	1,200
380B from scratch	4,600
Efficient Verification Strategy	**110**

Table 4: Comparison of computational costs across different verification strategies on a 1B LLM (transposed).

B Detailed Experimental Settings

Model Training Configuration. In our experiments, all models are trained using the open-source Megatron-LM library (Shoeybi et al., 2019). We utilize the MiniCPM-1.2B model architecture with the MiniCPM3-4B tokenizer. Each experiment involves training on 100B tokens (though the actual number is 104B tokens, calculated as $4096 \times 1024 \times 26000 = 104$B tokens; for simplicity, we refer to it as 100B), allowing for comprehensive data performance validation within computationally efficient parameters. Key training parameters include a sequence length of 4096, weight decay of 0.1, and a gradient clipping threshold of 1.0. We employ a global batch size of 1,024 across 26,000 training steps. The learning rate follows a cosine decay schedule, with a warm-up phase of 1,000 steps. The initial learning rate is set to 1e-5, the maximum learning rate to 1e-2, and the final learning rate to 1e-3. To enhance training stability, we use Maximal Update Parameterization (MuP) (Yang et al., 2022). Additionally, we save a checkpoint every 1,000 steps (approximately 4B tokens) for analysis during the training process. Detailed model configurations are provided in Table 5, where $Params.$, $Vocab.$, d_m, d_{ff}, d_h, n_{head}, n_{kv}, and n_{Layer} represent the total number of non-embedding parameters, vocabulary size, model hidden dimension, feedforward layer bottleneck dimension, attention head dimension, number of queries, number of key/values, and the number of layers, respectively.

Configuration	Value
Name	MiniCPM-1.2B
$Params.$	1,247,442,432
$Vocab.$	73,448
d_m	1,536
d_{ff}	3,840
d_h	64
n_{head}	24
n_{kv}	8
n_{Layer}	52

Table 5: Model Configurations for the MiniCPM-1.2B model (transposed).

Dataset Composition. We conduct two types of experiments for evaluating the datasets generated by our pipeline:

- **Individual Data Experiments:** We perform isolated training runs using single datasets, facilitating direct comparisons between differently processed data from identical sources. For English datasets, FineWeb is chosen as the source dataset, and comparisons are made with FineWeb-edu and Ultra-FineWeb-en. For Chinese datasets, Chinese FineWeb is selected with comparisons to Chinese FineWeb-edu-v2 and Ultra-FineWeb-zh. In the ablation studies, we primarily use individual data experiments for analysis.
- **Mixed Data Experiments:** Similar to the CCI3-HQ (Wang et al., 2024) experiment, we use a mix of 60% English data, 30% Chinese data, and 10% code data. The English-Chinese comparisons involve three dataset combinations: (1) FineWeb and Chinese FineWeb, (2) FineWeb-edu and Chinese FineWeb-edu-v2, and (3) Ultra-FineWeb-en and Ultra-FineWeb-zh. The code data is sourced exclusively from the StarCoder-v2 dataset (Lozhkov et al., 2024), maintaining consistent proportions across all experimental conditions.

Evaluation Metrics. We employ the Lighteval (Fourrier et al., 2023) library for model evaluation, mirroring the setup used with FineWeb (Penedo et al., 2024) and CCI3-HQ (Wang et al., 2024). All evaluation metrics are based on a zero-shot setting. The evaluation metrics include:

- $Average_{English}$: Average score across standard English metrics including MMLU (Hendrycks et al., 2020), ARC-C (Clark et al., 2018), ARC-E (Clark et al., 2018), CommonSenseQA (Talmor et al., 2018), HellaSwag (Zellers et al., 2019), OpenbookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), and Winogrande (Sakaguchi et al., 2021).
- $Average_{Chinese}$: Average score of Chinese metrics, including C-Eval (Huang et al., 2023) and CMMLU (Li et al., 2023).
- *Average*: The combined average score of all the above evaluation metrics.

C Results and Analysis of Efficient Verification

C.1 Overall Effectiveness of Efficient Verification

To verify the effectiveness of the efficient verification strategy, we use the FineWeb and FineWeb-edu datasets, training both on the efficient verification and from-scratch 100B token strategies, and compare the results. For evaluation, we use OpenCompass (Contributors, 2023) for the model trained with the efficient verification strategy, and Lighteval (Fourrier et al., 2023) for the model trained from scratch with 100B tokens. The experimental results are shown in Table 6.

We can observe that the efficient verification strategy exhibited consistent trends across multiple evaluation tasks when compared to the from-scratch 100B model. For example, in metrics like MMLU, ARC-E, ARC-C, and OpenbookQA, FineWeb-edu consistently outperformed FineWeb under both training paradigms. Similarly, for metrics such as HellaSwag, PIQA, SIQA, and Winogrande, FineWeb-edu showed performance degradation compared to FineWeb, regardless of training strategy. Overall, the efficient verification strategy quickly revealed the impact of the validation data on various evaluation dimensions and provided accurate feedback. This strategy significantly reduces computational resource requirements, enabling more efficient data quality assessment and optimization, ultimately enhancing model training effectiveness.

	Efficient Verification			100B From Scratch		
Metrics	FineWeb	FineWeb-edu	Diff.	FineWeb	FineWeb-edu	Diff.
MMLU	45.84	47.35	+1.51	28.84	31.80	+2.96
HellaSwag	57.72	56.99	-0.73	42.91	42.17	-0.74
ARC-C	38.98	39.66	+0.68	25.17	34.56	+9.39
ARC-E	57.67	59.08	+1.41	59.18	69.95	+10.77
PIQA	74.48	72.91	-1.57	73.29	72.14	-1.15
SIQA	43.55	43.35	-0.20	38.95	38.13	-0.82
Winogrande	56.67	55.56	-1.11	55.64	55.56	-0.08
OpenbookQA	66.80	69.40	+2.60	22.20	25.20	+3.00
Average	55.21	55.54	+0.33	41.00	43.00	+2.00

Table 6: Comparison of efficient verification strategy and from-scratch 100B token strategies on FineWeb and FineWeb-edu.

C.2 Stability under a Fixed Verification Ratio

In our experiments, we adopt a setting that allocates 30% of the training weight to the verification data. This choice was primarily motivated by the need to stabilize training and reduce the interference caused by fluctuations in optimization. To assess the stability of this setting, we conducted three independent runs using the same configuration. The results are shown in Table 7.

These results suggest that allocating 30% weight to verification data provides a stable and reliable training signal, enabling efficient verification to deliver consistent feedback with minimal sensitivity to stochastic training effects.

In the following subsection, we analyze the sensitivity of efficient verification to varying verification data weights.

C.3 Impact of Different Verification Ratios

To investigate the effect of different verification ratios, we conduct a two-stage annealing process with 10B tokens on the FineWeb, FineWeb-edu, and Ultra-FineWeb-en datasets. We evaluate six distinct verification ratios: 5%, 10%, 20%, 30%, 40%, and 50%. The comprehensive results are presented in Table 8.

From a stability standpoint, overall performance trends remain consistent across different verification ratios. However, at lower ratios such as 5% and 10%, the performance gains from verification data are marginal and fail to clearly distinguish the quality differences across datasets. Starting from 20%, performance variances across benchmarks become more pronounced. For instance, compared to FineWeb, the MMLU performance improvement of Ultra-FineWeb-en is +1.22pp at 20%, peaking at +2.18pp at 30% and +2.16pp at 40%. Notably, when the verification ratio is further increased to 50%, the improvement diminishes to +0.87pp. This indicates that a higher verification ratio does not linearly yield better results and may introduce issues such as sample redundancy or distributional skew, thereby complicating the interpretability of the results.

Moreover, the choice of verification ratio influences the discernibility of evaluation results. Our strategy demonstrates strong generalizability, allowing practitioners to dynamically adjust verification ratios according to their specific training configurations. By conducting a small number of full-scale pilot experiments, researchers can establish mappings between datasets and determine appropriate evaluation thresholds, enabling more stable and reliable assessments. Additionally, this mapping mechanism can support the selection of high-quality data tailored to specific evaluation benchmarks.

Therefore, the proposed **Efficient Verification Strategy** offers an efficient and robust means of evaluating pre-training data quality with controlled computational costs. These findings substantiate the selection of verification ratios and confirm the robustness of our experimental framework.

C.4 Model-Agnostic Behavior across Training Cutoffs

To further demonstrate the generality of efficient verification, we further evaluate whether the conclusions drawn from efficient verification remain consistent across different pre-training cutoffs and token budgets. Given the constraints of computational resources, we selected two additional representative settings to compare against our main experiments:

1. **Jan. 2025 Cutoff**: A 1B model trained on

Metrics	Baseline	Run 1	Run 2	Run 3	Mean	Std
MMLU	46.75	47.35	47.51	47.29	47.38	0.013
HellaSwag	55.72	56.99	56.99	56.86	56.95	0.006
ARC-C	38.31	39.66	38.58	38.64	38.96	0.368
ARC-E	57.32	59.08	59.30	58.91	59.10	0.038
PIQA	72.74	72.91	72.95	73.01	72.96	0.003
SIQA	41.97	43.35	43.09	43.07	43.17	0.024
Winogrande	52.46	55.56	55.63	55.91	55.70	0.034
OpenbookQA	70.40	69.40	69.00	69.40	69.27	0.053

Table 7: Stability analysis of efficient verification with 30% verification data weight. Results are reported over three independent runs under identical settings.

Ratio	Dataset	MMLU	HellaS.	ARC-C	ARC-E	PIQA	SIQA	Wino.	OBQA	Average
-	**Baseline**	46.75	55.72	38.31	57.32	72.74	41.97	52.46	70.40	54.46
5%	FW	46.71	56.38	39.02	57.91	72.99	42.78	52.78	70.40	54.87
	FE	$46.79_{+0.08}$	$56.02_{-0.36}$	$39.08_{+0.06}$	$58.55_{+0.64}$	$73.03_{+0.04}$	$42.57_{-0.21}$	$52.69_{-0.09}$	$70.60_{+0.20}$	$54.92_{+0.05}$
	UFW_{en}	$46.95_{+0.24}$	$56.03_{-0.35}$	$39.21_{+0.19}$	$58.73_{+0.82}$	$73.09_{+0.10}$	$42.93_{+0.15}$	$53.06_{+0.28}$	$70.80_{+0.40}$	$55.10_{+0.23}$
10%	FW	46.73	56.89	38.31	59.08	73.83	42.43	55.33	69.60	55.28
	FE	$46.93_{+0.20}$	$56.26_{-0.63}$	$38.64_{+0.33}$	$60.14_{+1.06}$	$73.12_{-0.71}$	$42.07_{-0.36}$	$54.99_{-0.34}$	$70.60_{+1.00}$	$55.34_{+0.07}$
	UFW_{en}	$47.05_{+0.32}$	$56.35_{-0.54}$	$39.02_{+0.71}$	$60.14_{+1.06}$	$73.91_{+0.08}$	$42.84_{+0.41}$	$55.65_{+0.32}$	$70.60_{+1.00}$	$55.70_{+0.42}$
20%	FW	46.27	57.45	38.64	57.50	73.39	42.99	56.75	69.00	55.25
	FE	$47.31_{+1.04}$	$56.57_{-0.88}$	$39.27_{+0.63}$	$58.51_{+1.01}$	$72.31_{-1.08}$	$42.82_{-0.17}$	$55.68_{-1.07}$	$71.20_{+2.20}$	$55.46_{+0.21}$
	UFW_{en}	$47.49_{+1.22}$	$56.83_{-0.62}$	$39.29_{+0.65}$	$59.61_{+2.11}$	$73.45_{+0.06}$	$43.14_{+0.15}$	$56.82_{+0.07}$	$71.80_{+2.80}$	$56.05_{+0.81}$
30%	FW	45.84	57.72	38.98	57.67	74.48	43.55	56.67	66.80	55.21
	FE	$47.35_{+1.51}$	$56.99_{-0.73}$	$39.66_{+0.68}$	$59.08_{+1.41}$	$72.91_{-1.57}$	$43.35_{-0.20}$	$55.56_{-1.11}$	$69.40_{+2.60}$	$55.54_{+0.32}$
	UFW_{en}	$48.02_{+2.18}$	$57.10_{-0.62}$	$39.70_{+0.72}$	$60.32_{+2.65}$	$74.23_{-0.25}$	$44.06_{+0.51}$	$56.79_{+0.12}$	$70.20_{+3.40}$	$56.30_{+1.09}$
40%	FW	45.96	58.19	39.02	58.73	74.27	43.24	58.07	66.20	55.46
	FE	$47.27_{+1.31}$	$57.33_{-0.86}$	$39.54_{+0.52}$	$59.79_{+1.06}$	$73.12_{-1.15}$	$42.89_{-0.35}$	$57.24_{-0.83}$	$68.40_{+2.20}$	$55.70_{+0.24}$
	UFW_{en}	$48.12_{+2.16}$	$57.86_{-0.33}$	$39.72_{+0.70}$	$60.54_{+1.81}$	$73.78_{-0.49}$	$43.34_{+0.10}$	$58.05_{-0.02}$	$69.40_{+3.20}$	$56.35_{+0.89}$
50%	FW	45.50	58.17	38.68	58.01	73.94	42.94	57.85	66.20	55.16
	FE	$46.14_{+0.64}$	$57.23_{-0.94}$	$39.64_{+0.96}$	$59.61_{+1.60}$	$73.39_{-0.55}$	$43.05_{+0.11}$	$56.49_{-1.36}$	$67.40_{+1.20}$	$55.37_{+0.21}$
	UFW_{en}	$46.37_{+0.87}$	$57.78_{-0.39}$	$40.00_{+1.32}$	$60.85_{+2.84}$	$73.61_{-0.33}$	$44.06_{+1.12}$	$57.90_{+0.05}$	$68.20_{+2.00}$	$56.10_{+0.94}$

Table 8: Performance comparison across different verification ratios. Base denotes the baseline. Subscripts represent the performance delta (Δ) relative to the baseline. We use **FW**, **FE**, and $\textbf{UFW}_{en}$ to denote FineWeb, FineWeb-edu, and Ultra-FineWeb-en, respectively.

1.1T tokens (1T stable phase, 100B decay phase).

2. **Jun. 2025 Cutoff**: A 1B model trained on 1.5T tokens (1.3T stable phase, 200B decay phase).

The comparative results across three distinct data snapshots (Sep. 2024, Jan. 2025, and Jun. 2025) are detailed in Table 9.

Experimental results indicate that as the pre-training cutoff progresses and total token count increases, the absolute performance of the model improves across most benchmarks. Crucially, the *relative ranking* of verification datasets remains consistent: Ultra-FineWeb-en consistently outperforms FineWeb-edu, which in turn surpasses the baseline FineWeb. This stability underscores the robust generalizability of our proposed verification strategy across different stages of model maturity.

However, we observe that the performance delta (Δ) between datasets tends to narrow as the pre-training data quality and volume increase. For instance, in the Sep. 2024 setting, Ultra-FineWeb-en achieves a +1.09pp gain over FineWeb, whereas this advantage reduces to +0.52pp by the Jun. 2025 cutoff. We attribute this phenomenon to two factors: first, a potential performance plateau as models approach their capacity on specific benchmarks; and second, the likelihood that high-quality corpora (similar to Ultra-FineWeb-en) were already integrated into the larger pre-training mixes used for the 2025 models, thereby partially saturating the marginal gains during the annealing phase.

In summary, the proposed efficient verification strategy remains a reliable proxy for data quality assessment, independent of the training cutoff. It enables practitioners to establish stable mappings between datasets and downstream performance, fa-

Metric	Sep. 2024 Cutoff (Main)			Jan. 2025 Cutoff			Jun. 2025 Cutoff		
	FW	FE	UFW_{en}	FW	FE	UFW_{en}	FW	FE	UFW_{en}
MMLU	45.84	$47.35_{+1.51}$	$48.02_{+2.18}$	51.13	$52.45_{+1.32}$	$53.04_{+1.91}$	53.61	$53.72_{+0.11}$	$53.86_{+0.25}$
HellaSwag	57.72	$56.99_{-0.73}$	$57.10_{-0.62}$	58.64	$58.42_{-0.22}$	$58.64_{+0.00}$	57.71	$56.68_{-1.03}$	$57.04_{-0.67}$
ARC-C	38.98	$39.66_{+0.68}$	$39.70_{+0.72}$	38.31	$38.64_{+0.33}$	$39.32_{+1.01}$	39.66	$40.00_{+0.34}$	$39.88_{+0.22}$
ARC-E	57.67	$59.08_{+1.41}$	$60.32_{+2.65}$	62.08	$63.32_{+1.24}$	$64.02_{+1.94}$	58.55	$59.79_{+1.24}$	$60.67_{+2.12}$
PIQA	74.48	$72.91_{-1.57}$	$74.23_{-0.25}$	75.08	$74.94_{-0.14}$	$75.14_{+0.06}$	73.88	$73.86_{-0.02}$	$74.16_{+0.28}$
SIQA	43.55	$43.35_{-0.20}$	$44.06_{+0.51}$	44.32	$44.28_{-0.04}$	$44.55_{+0.23}$	43.19	$43.20_{+0.01}$	$43.81_{+0.62}$
Winogrande	56.67	$55.56_{-1.11}$	$56.79_{+0.12}$	57.06	$56.62_{-0.44}$	$57.10_{+0.04}$	56.91	$56.90_{-0.01}$	$57.03_{+0.12}$
OpenbookQA	66.80	$69.40_{+2.60}$	$70.20_{+3.40}$	70.00	$71.40_{+1.40}$	$71.80_{+1.80}$	74.60	$74.80_{+0.20}$	$75.80_{+1.20}$
Average	55.21	$55.54_{+0.32}$	**56.30**$_{+1.09}$	57.08	$57.51_{+0.43}$	**57.95**$_{+0.87}$	57.26	$57.37_{+0.11}$	**57.78**$_{+0.52}$

Table 9: Performance comparison across different training data cutoffs and token budgets. Subscripts denote the performance delta (Δ) relative to the FineWeb (FW) baseline for each respective period. FE and UFW_{en} represent FineWeb-edu and Ultra-FineWeb-en, respectively.

cilitating informed decisions for data filtering and selection even as baseline model performance continues to evolve.

D Further Analysis and Ablation Studies

D.1 Classifier Inference Intersection Results.

To investigate the impact of intersecting positive samples from multiple classifiers on LLM performance, we conduct experiments using the intersection of classifier-inferred positive samples for model training. As demonstrated in Tables 10, the model trained on Ultra-FineWeb-en_{inter} exhibits substantial performance gains across multiple English metrics. Compared to the Ultra-FineWeb-en, the score improved by 0.447pp, with the most significant improvements observed in tasks such as MMLU, ARC-C, ARC-E, and OpenbookQA. Similarly, for Chinese metrics, the model trained on Ultra-FineWeb-zh_{inter} also showed notable performance gains, with the overall Chinese average score increasing from 35.16pp to 36.455pp. In particular, the score in CMMLU improved by 1.8pp compared to Ultra-FineWeb-zh. Additionally, we visualize the evaluation scores at each checkpoint during training, as shown in Figure 4, where the model using intersection data consistently maintain the highest score throughout training. These results indicate that combining the inference results from multiple classifiers, particularly through intersecting positive sample data, can significantly further enhance model performance, yielding significant improvements across key metrics.

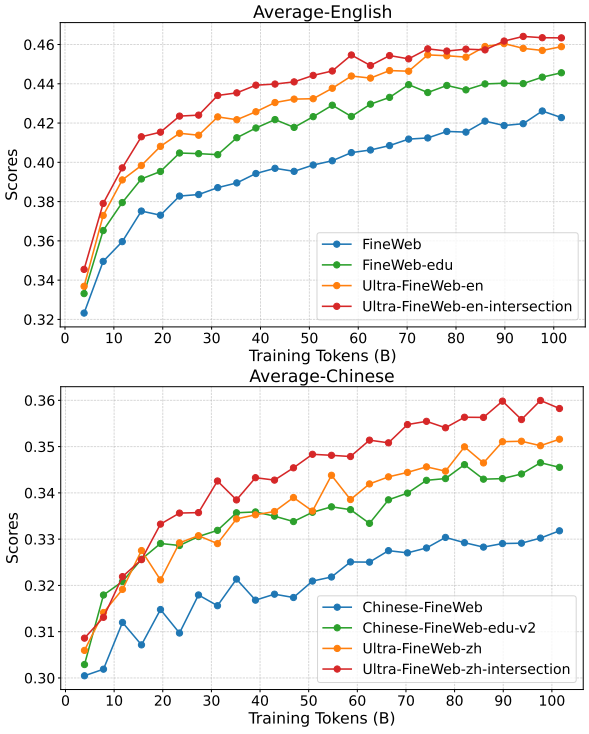


Figure 4: **Average scores at each checkpoint during training for different datasets**: Compared with using the intersection of positive samples inferred by multiple classifiers.

D.2 Analysis of Token Length Distributions

Inspired by ProX (Zhou et al., 2024), we analyze the token length distributions across different datasets, as shown in Figure 5. For the English

Metrics	FineWeb	FineWeb-edu	Ultra-FineWeb-en	Ultra-FineWeb-en$_{inter}$
MMLU	28.84	31.80$_{+2.96}$	32.24$_{+3.40}$	**33.37**$_{+4.53}$
ARC-C	25.17	34.56$_{+9.39}$	35.67$_{+10.50}$	**38.31**$_{+13.14}$
ARC-E	59.18	69.95$_{+10.77}$	70.62$_{+11.44}$	**73.48**$_{+14.30}$
CommonSenseQA	34.32	31.53$_{-2.79}$	36.45$_{+2.13}$	**36.94**$_{+2.62}$
HellaSwag	**42.91**	42.17$_{-0.74}$	42.76$_{-0.15}$	41.39$_{-1.52}$
OpenbookQA	22.20	25.20$_{+3.00}$	26.20$_{+4.00}$	**28.60**$_{+6.40}$
PIQA	73.29	72.14$_{-1.15}$	**73.67**$_{+0.38}$	71.16$_{-2.13}$
SIQA	38.95	38.13$_{-0.82}$	**39.61**$_{+0.66}$	39.41$_{+0.46}$
Winogrande	55.64	55.56$_{-0.08}$	**55.80**$_{+0.16}$	54.38$_{-1.26}$
Average$_{English}$	42.278	44.560$_{+2.282}$	45.891$_{+3.613}$	**46.338**$_{+4.06}$
Metrics	Chinese-FineWeb	Chinese-FineWeb-edu-v2	Ultra-FineWeb-zh	Ultra-FineWeb-zh$_{inter}$
C-Eval	33.95	34.17$_{+0.22}$	34.26$_{+0.31}$	**35.05**$_{+1.1}$
CMMLU	32.41	34.93$_{+2.52}$	36.06$_{+3.65}$	**37.86**$_{+5.45}$
Average$_{Chinese}$	33.180	34.550$_{+1.37}$	35.160$_{+1.98}$	**36.455**$_{+3.275}$

Table 10: Comparison of results on English and Chinese datasets using the intersection of positive samples inferred by multiple classifiers.

datasets, the token length distributions of Ultra-FineWeb-en and FineWeb are quite similar, while FineWeb-edu exhibits a rightward shift, indicating that the classifier tends to extract longer tokens. In terms of average token length, FineWeb has the shortest average, followed by Ultra-FineWeb, with FineWeb-edu having the longest average token length. For the Chinese datasets, Ultra-FineWeb-zh and Chinese FineWeb exhibit similar token length distributions, while Chinese FineWeb-edu-v2 also shows a rightward shift. The average token length follows the order: Chinese FineWeb < Chinese FineWeb-edu-v2 < Ultra-FineWeb-zh. We believe these differences may stem from the inherent preference of LLM-based models, which tend to favor longer tokens in their scoring. Additionally, this phenomenon might be further influenced by training recipes, as LLM-based models label data from the same source, typically assigning lower scores to shorter texts and higher scores to longer ones, leading classifiers to favor longer texts. In contrast, our data seeds are more diverse, making the classifiers less focused on token length, which results in the token length distribution of our extracted data aligning more closely with the original source data.

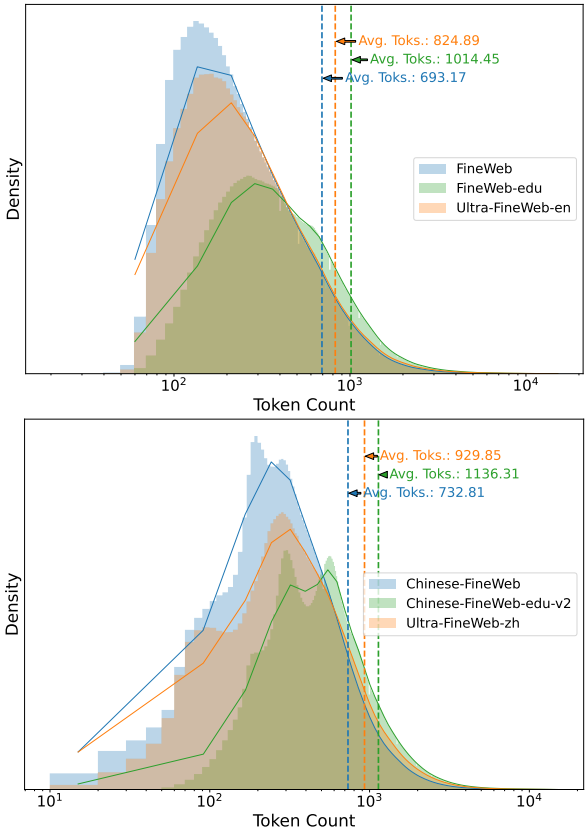


Figure 5: Comparison of token length distributions across different datasets.

D.3 Loss and Performance Estimation Results

We use the performance estimation methods proposed in (Xiao et al., 2024) for further analysis and verification of the effectiveness of Ultra-FineWeb. First, we establish the standard configuration in (Xiao et al., 2024) as the baseline. Specif-

ically, we adopt the MiniCPM-3-4B (Hu et al., 2024) training corpus, applying models across six scales (0.005B, 0.03B, 0.1B, 0.2B, 0.4B, 0.8B), and train with six token configurations (10, 15, 20, 30, 40, 60 × N, where N represents the model parameter size). Based on these 36 models, we compute and plot the compute ($= 6ND$)-Loss curve, and subsequently predict the performance of each model using the Loss-Performance curve from the Densing Law. This analysis is performed on MMLU (Hendrycks et al., 2020), BBH (Suzgun et al., 2022), MATH (Hendrycks et al., 2021), MBPP (Austin et al., 2021), HumanEval (Chen et al., 2021), C-Eval (Huang et al., 2023), and CMMLU (Li et al., 2023) evaluation metrics. Next, we replace the "High-Quality" data in the baseline with ***Ultra-FineWeb*** and repeat the experiment, performing Loss Estimation. Finally, through this two-step estimation, we predict the performance of an 8B model trained on 8T tokens. The loss values and estimated results are shown in Table 11, with the Loss-Performance curve shown in Figure 6. Experimental results demonstrate that using Ultra-FineWeb significantly reduces the loss for metrics such as MMLU, MATH, C-Eval, and CMMLU, thereby improves model performance.

D.4 Ablation Study on Multi-Source Seed Selection

To verify the impact of selecting multi-source seed on the robustness of the classifier during the efficient data filtering pipeline process, we choose DCLM-Pool (Li et al., 2024) as the English data source and MAP-CC (Du et al., 2024) as the Chinese data source for verification. In the experiment, we compare the performance of models trained with original data, LLM-based classifier (-edu), and data extracted by our classifier (Ultra-) on different evaluation sets. Notably, due to the unavailability of an open-source LLM-based classifier for Chinese-FineWeb-edu, we only compare the performance difference between the original MAP-CC data and the data extracted by our classifier (Ultra-MAP-CC). As detailed in Tables 12 and 13, Ultra-DCLM demonstrates superior performance over both DCLM-Pool and DCLM-edu across multiple English evaluation tasks. The English average score for Ultra-DCLM (47.252pp) shows a 1.671pp improvement over DCLM-Pool (45.581pp) and a 0.658pp advantage over DCLM-edu (46.594pp), with particularly notable gains in MMLU, ARC-C, and OpenbookQA metrics. For Chinese evaluations, Ultra-MAP-CC also exhibits significant enhancements, especially in CMMLU with a 2.8pp increase, achieving an overall 1.43pp improvement over the original dataset. These results demonstrate that our classifier remains highly robust and effective even in non-homogeneous data scenarios, further confirming the positive impact of the multi-source seed selection strategy on improving classifier robustness and performance. Figure 7 presents the evaluation results at each checkpoint during training. In the early stages of training, the performance of Ultra-DCLM and DCLM-edu is similar, but both outperform DCLM-Pool significantly. When training reaches 30B tokens, Ultra-DCLM begins to surpass DCLM-edu. For the Chinese evaluation sets, Ultra-MAP-CC significantly outperforms MAP-CC from the early stages of training.

E Use of AI

We used ChatGPT to support language polishing of the manuscript. The tool was applied to improve grammar, fluency, and stylistic consistency in English writing. It was not used for generating research ideas, experimental designs, datasets, models, results, or interpretations. All technical content and scientific contributions are solely the work of the authors.

Metrics	Baseline		Ultra-FineWeb	
	Loss	Estimate Acc.	Loss	Estimate Acc.
MMLU	0.182	70.84	0.143$_{-0.039}$	**85.60**$_{+14.76}$
BBH	0.097	56.70	0.092$_{-0.005}$	**60.48**$_{+3.78}$
MATH	0.225	25.96	0.162$_{-0.063}$	**59.05**$_{+33.09}$
MBPP	0.175	**84.91**	0.176$_{+0.001}$	84.87$_{-0.04}$
HumanEval	0.119	48.18	0.113$_{-0.006}$	**54.81**$_{+6.63}$
C-Eval	0.244	60.44	0.226$_{-0.018}$	**69.33**$_{+8.89}$
CMMLU	0.243	66.02	0.226$_{-0.017}$	**73.75**$_{+7.73}$
Average	0.189	42.40	0.174$_{-0.015}$	**49.85**$_{+7.45}$

Table 11: Loss values and estimated performance for 8B model trained on 8T tokens.

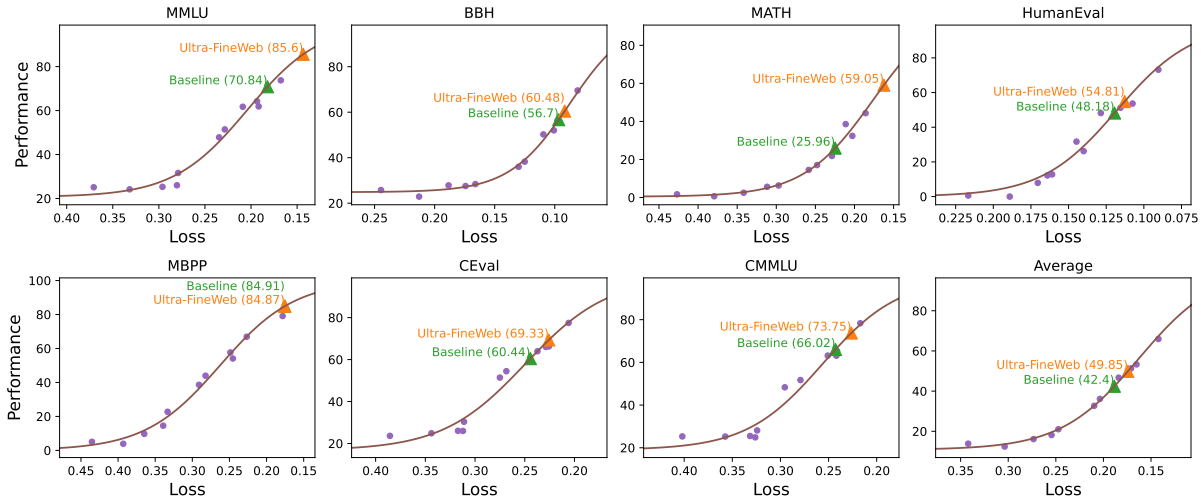


Figure 6: **Loss-performance curve:** Showing the estimated performance of an 8B model trained on 8T tokens using baseline and replacing high-quality data with Ultra-FineWeb.

Metrics	DCLM-Pool	DCLM-edu	Ultra-DCLM
MMLU	31.45	34.07$_{+2.62}$	**34.33**$_{+2.88}$
ARC-C	31.48	37.71$_{+6.23}$	**38.48**$_{+7.00}$
ARC-E	66.08	**73.40**$_{+7.32}$	72.77$_{+6.69}$
CommonSenseQA	**41.52**	39.72$_{-1.80}$	40.70$_{-0.82}$
HellaSwag	**44.28**	41.77$_{-2.51}$	43.31$_{-0.97}$
OpenbookQA	25.00	26.60$_{+1.60}$	**27.40**$_{+2.40}$
PIQA	73.67	70.73$_{-2.94}$	**73.89**$_{+0.22}$
SIQA	**40.79**	39.00$_{-1.79}$	39.51$_{-1.28}$
Winogrande	55.96	**56.35**$_{+0.39}$	54.88$_{-1.08}$
$Average_{English}$	45.581	46.594$_{+1.013}$	**47.252**$_{+1.671}$

Table 12: Comparison of results on DCLM-Pool-based datasets.

Metrics	MAP-CC	Ultra-MAP-CC
C-Eval	34.58	**34.64**$_{+0.06}$
CMMLU	32.02	**34.82**$_{+2.80}$
$Average_{Chinese}$	33.300	**34.730**$_{+1.430}$

Table 13: Comparison of results on MAP-CC-based datasets.

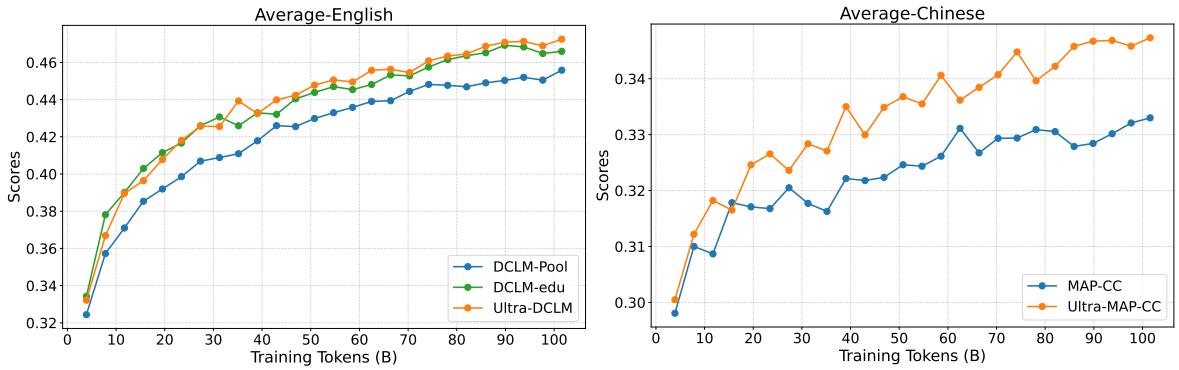


Figure 7: Average scores at each checkpoint during training for different source data.