PCMind-2.1-Kaiyuan-2B Technical Report

Kairong Luo[1]*, Zhenbo Sun[1]*, Xinyu Shi[1], Shengqi Chen[1], Bowen Yu[3], Yunyi Chen[1], Chenyi Dang[1], Hengtao Tao[2], Hui Wang[2], Fangming Liu[2], Kaifeng Lyu[1], Wenguang Chen[1,2†]
[1]Tsinghua University, [2]Peng Cheng Laboratory, [3]Beijing Houdu Technology Co., Ltd

Abstract

The rapid advancement of Large Language Models (LLMs) has resulted in a critical knowledge gap between the open-source community and industry, primarily because the latter relies on closed-source, high-quality data and training recipes. To address this, we introduce **PCMind-2.1-Kaiyuan-2B (Kaiyuan-2B)**, a fully open-source 2-billion-parameter model focused on improving training efficiency and effectiveness under resource constraints. Our methodology introduces three key innovations: a *Quantile Data Benchmarking* method for systematically comparing heterogeneous open-source datasets and providing insights on how to mix them; a *Strategic Manual Repetition* scheme within a multi-phase paradigm to effectively leverage sparse, high-quality data; and a *Multi-Domain Curriculum Training* policy that orders samples by quality. Supported by a highly optimized data preprocessing pipeline and architectural modifications for FP16 stability, Kaiyuan-2B achieves performance competitive with state-of-the-art fully open-source models, demonstrating practical and scalable solutions for resource-limited pretraining. The HuggingFace link for open-source assets is `https://huggingface.co/thu-pacman/PCMind-2.1-Kaiyuan-2B`.

1 Introduction

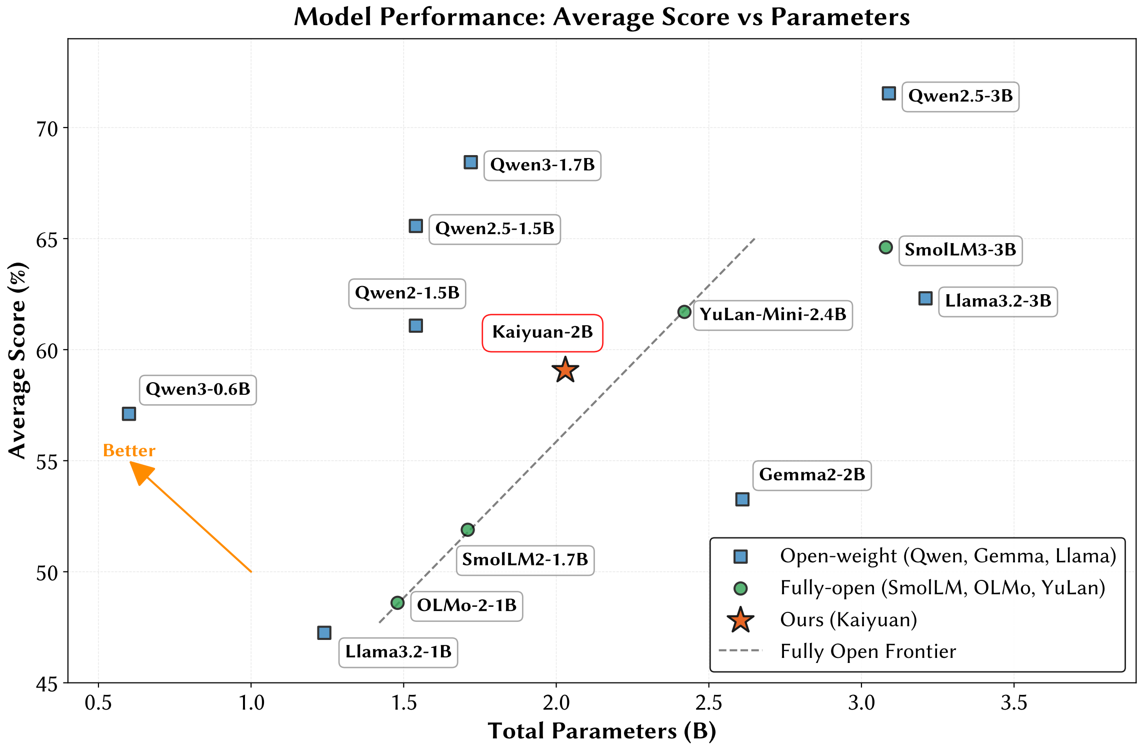


Figure 1: Model performance comparison. Kaiyuan-2B surpasses the frontier of fully open-source models at a similar scale, and closely approaches open-weight models such as Qwen2-1.5B [77] and Llama3.2-3B [50]. A full version of the corresponding benchmark scores is detailed in Table 17.

*luokr24@mails.tsinghua.edu.cn, sunzb20@mails.tsinghua.edu.cn
†Corresponding authors.

The field of Large Language Models (LLMs) has seen remarkable advancements, demonstrating comprehensive capabilities across a wide spectrum of tasks. The performance of these models fundamentally depends on the quality and scale of their pretraining [83, 34]. However, the core science and engineering behind large-scale LLM pretraining remain underexplored by the academic and open-source communities due to two main industry practices:

1. **Closed-source pretraining** of leading models [58, 70].
2. Release of **open model weights** but with **closed-source training recipes** [21, 20, 76].

Fully open-source models, which publish both weights, datasets, and detailed pretraining procedures, are essential to bridge this knowledge gap and facilitate academic exploration. Pioneer works in this direction include the OLMo series [24, 57, 53], SmolLM series [1, 7], and Yulan series [92, 36]. Furthermore, the increasing availability of high-quality, open-source pretraining datasets, spanning English, multilingual, code, and math domains [61, 43, 46, 39, 17, 81, 90], lays a crucial foundation for more open pretraining attempts.

Despite these advancements, significant challenges persist in open-source pretraining, particularly when attempting to match the performance of state-of-the-art open-weight models under resource constraints. Our work focuses on two critical challenges faced by resource-limited communities:

1. **Heterogeneous Open-Source Data:** While many pretraining-scale datasets are available, their sources and preprocessing pipelines vary significantly [43, 61, 46]. This leads to vast differences in data features, posing a challenge for the effective comparison, selection, and mixing of these heterogeneous datasets [67].
2. **Limited Compute Resources:** The academic community typically cannot afford to train on the scale of tokens (e.g., tens of trillions) used by industry leaders [76]. This necessitates novel strategies to improve training efficiency with limited data and computational resources.

In this technical report, we introduce the **PCMind-2.1-Kaiyuan-2B (Kaiyuan-2B), fully open-source model**, and detail its pretraining methodology. Our primary goal is to push the frontier of open-source pretraining by directly addressing these two questions:

1. How can one properly compare, select, and effectively mix heterogeneous open-source datasets?
2. How to improve the training efficiency, especially when dealing with the inherent sparsity of high-quality data?

In the pretraining of the Kaiyuan-2B model, we propose and implement practical solutions to these challenges, centered on data management and training efficiency. As shown in Figure 1, through these practices, our Kaiyuan-2B model achieves competitive performance among fully open-source models at a similar scale, even approaching open-weight models like Qwen2-1.5B [77] and Llama3.2-3B [50].

Deduplication and Quantile Data Benchmarking. We propose a novel **quantile benchmarking** method to systematically evaluate and compare leading open-source datasets (e.g., DCLM Baseline [43], Fineweb-Edu [61]). The rationale is twofold: (1) Open-source datasets often include rule-based or model-based quality metrics, which have proven effective in filtering and can be used to inform the importance of samples during comparison [61, 43]. (2) By selecting a data subset around a target quality score quantile and training a small reference model over it, we can measure the subset's characteristics via the reference model's downstream performance. This method allows us to understand how different datasets or distinct partitions within a single dataset perform across various capabilities, enabling systematic benchmarking across heterogeneous collections, especially for the leading datasets that account for the majority of training tokens. Crucially, deduplication is performed before the quantile benchmarking process (Section 3).

Strategic Manual Repetition for Sparse High-Quality Data. Our data benchmarking confirms that high-quality data is extremely useful but sparse. To exploit this utility without excessive resource expenditure, we adopt a multi-phase training paradigm that implements manual repetition. Specifically, one dataset can occur in multiple phases rather than appear only once in the whole training process. However, in each phase, each data sample mostly occurs only once. Moreover, instead of repeating the whole dataset, we mostly keep only the high-quality partition and retain fewer topmost samples in the latter phases when the quality metrics are available. This ensures that higher-quality data samples are repeated more frequently. We employ a five-phase training pipeline, which limits repetition such that the overall benefit remains similar to that observed in one-pass training regimes [52, 75].

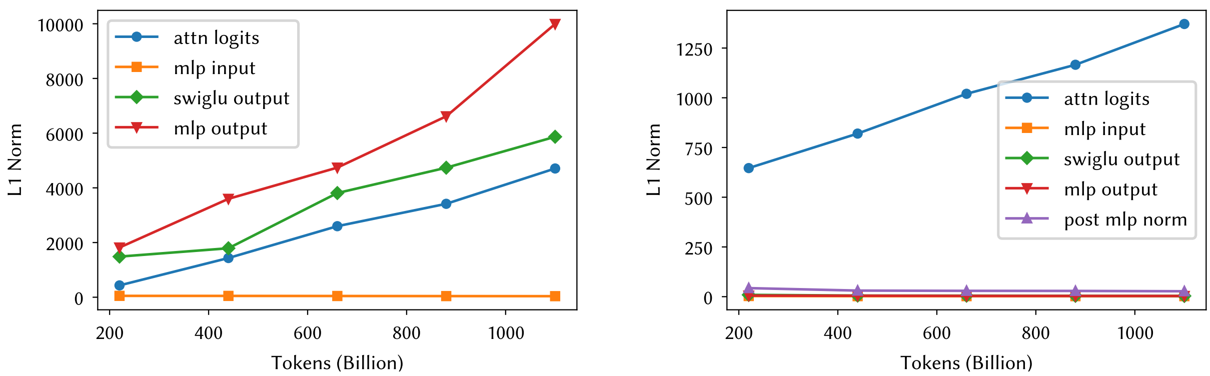
Multi-Domain Curriculum Training. In addition to strategic repetition, we integrate a data curriculum within training phases 3, 4, and 5. This curriculum ensures a stable data mixture across different datasets while sorting data samples in ascending order of their quality metrics within each dataset. Datasets without explicit quality labels are simply shuffled. This means that more important and high-quality samples are presented to the model in the latter training steps. To make full use of the benefit of the data curriculum, we adopt a moderate Learning Rate (LR) decay and apply model averaging over the last several checkpoints, following recent findings [48].

System Infrastructure and Training Stability. To support these data-centric efforts, we built a high-performance and scalable data preprocessing pipeline based on Spark [84] and optimized with the Chukonu framework [80]. This optimized framework efficiently supports deduplication and leverages Spark's native sorting for curriculum implementation. Finally, our pretraining experiments were conducted on Ascend 910A clusters. Ascend 910A is comparable to V100 hardware and supports only FP16. To ensure training stability under these conditions, we modify the model architecture based on Qwen3-1.7B by incorporating sandwich normalization and soft capping, in addition to standard QK normalization.

In summary, the Kaiyuan-2B project delivers a fully open-source pretraining attempt, accompanied by an open-source data preprocessing framework, the final pretraining dataset, and the model checkpoint. Our core contributions lie in the practical exploration of dataset benchmarking and the design of strategic repetition and curriculum training policies. We hope that Kaiyuan-2B will serve as a valuable resource and contribute to the advancement of the open-source LLM community.

The rest of this report is organized as follows: In Section 2, we will discuss how to stabilize training on FP16-only hardware through architecture design. In Section 3, we will introduce our quantile benchmarking approach to deepen our understanding of how various score metrics reflect the data inherent in different feature dimensions. In Section 4, we will discuss two approaches to leverage high-quality data in our training: selective repetition and quality-based curriculum. Then in Section 5, we will report our evaluation settings and results, positioning Kaiyuan-2B in the fully-open and open-weight models. Additionally, Section A shows model performance comparison relative to non-embedding parameters; Section B lists quantile benchmarking results; Section C lists all used datasets along with license details; Section D lists dataset mixture details in all phases; Section E presents the implementation details and experiment settings in our training and small-scale experiments. Section F shows the full table for model performance comparison.

2 Architecture Design and Training Stability



(a) Activation statistics of the baseline architecture

(b) Activation statistics after applying Sandwich Normalization and Logits Soft-capping

Figure 2: Comparison of internal activation magnitudes before and after architectural optimization. The experiment is conducted with a 3B model.

Kaiyuan-2B is trained on Huawei Ascend 910A accelerators. Similar to NVIDIA V100s, these devices rely on FP16 precision to achieve high training efficiency. However, FP16 has a limited dynamic numerical range, which introduces overflow risks when model parameters or activations grow too large. To keep training stable, we first identify the activations that are most likely to overflow and then introduce structural changes that keep their values within safe bounds.

Following the standard Llama architecture, the model uses SwiGLU [19], RMSNorm [86], and RoPE [68]. We adopt mixed precision training, where operators that need higher precision, such as Softmax and RMSNorm, run in FP32, and the remaining computations run in FP16. Despite this setup, training on large and diverse datasets, including code and mathematics, still leads to strong numerical instability. As shown in Figure 2a, most instability comes from two places: the attention logits and the activations after the SwiGLU function in the MLP layers. In practice, the maximum activation values grow without control. They exceed 10,000 after processing one trillion tokens, which is close to the FP16 upper limit. As a result, the dynamic loss scaler decreases its scaling factor to avoid overflow. This drop pushes many gradients below the FP16 minimum representable value, which causes underflow. The gradients then become inaccurate, harming convergence and sometimes causing training to fail.

To solve these issues, we use Logits Soft-Capping [8] and Sandwich Normalization [22]. This follows the design choices of Gemma 2 [62]. These techniques place strict bounds on activation values. As shown in Figure 2b, soft-capping reduces the L1 norm of attention logits by about an order of magnitude. At the same time, sandwich normalization reduces the accumulation of large values in residual connections and keeps the L1 norm of MLP activations within a safe range. To further improve stability, we set the weight decay to 0.1, apply soft-capping to the final output logits, and replace the soft-capping inside each attention layer with QK-Norm [33]. The full configuration of Kaiyuan-2B is listed in Table 11 and the implementation details are discussed in Section E.1.

3 Data Benchmarking and Preprocessing

There are many open-source pretraining datasets across various data domains, especially for English, Code, and Math [57, 1, 79, 43, 67, 46, 90]. However, constructing a high-quality pretraining corpus remains a non-trivial task due to two primary challenges.

First, it is different to measure the quality of diverse datasets and determine the optimal strategy for selecting and mixing data from heterogeneous sources. Second, preprocessing these datasets is both resource-intensive and technically complex. Given the large scale of pretraining data and complex operations like deduplication, the preprocessing pipeline incurs substantial computational overhead and engineering complexity.

To mitigate these issues, (1) we propose to benchmark primary datasets (e.g., DCLM Baseline [43], Fineweb-Edu [61]) by quantiles of quality scores. We train reference models over the data subsets around a series of quantiles of quality scores, and then become aware of how the resulting benchmark performance varies with data distribution, which is reflected by quality scores. (2) We develop a user-friendly Spark-based data preprocessing framework to efficiently process large-scale pretraining datasets. Moreover, we exploit the Chukonu [80] framework to reduce the preprocess overhead. These explorations on data dimensions lay a solid foundation for our training and future work.

3.1 Benchmarking Dataset By Quality-Score Quantiles

Background and Motivation. Most open-source datasets have been through a preprocessing pipeline, which primarily incorporates steps of quality scoring and data filtering by score. These score labels are typically released for these open-source datasets. Therefore, we can select a (hopefully) higher-quality subset based on sample quality scores. However, samples between different datasets are hard to compare, considering heterogeneous quality metrics. When scorers are available, it is possible to score both datasets. But as more datasets and quality metrics are included, it is hard to scale up and judge by multiple quality metrics. DCLM [43] proposes to benchmark datasets or quality scores by filtering and feeding datasets into a standard series of models across scales. However, when facing a practical pretraining setup, the top-k filtering and multi-scale benchmarking can be challenging: the cost of full benchmarking is prohibitive, and we need to ablate over different filtering ratios to balance quality and quantity of the filtered dataset.

Method and Implementation. Instead of relying solely on top-k filtering, we design a small-scale evaluation process across a range of quality score quantiles to benchmark dataset quality. In practice, preprocessed open-source datasets typically provide a quality metric for each data sample. These quality scores reflect specific characteristics of the data and can vary significantly. It is commonly expected that using higher-quality data samples in training should lead to better model performance. Motivated by this intuition, we propose a straightforward approach, yet not explicitly reported in previous work.

For a target dataset, we first determine a series of quality score quantiles, such as top 0%, 10%, 20%, . . . , 80%. At each quantile, we select a fixed-size subset of the dataset. In our implementation, starting from the data sample ranked at the top 10% in terms of quality score, we expand the subset by including the next 10B tokens of lower-quality samples to form the probing dataset. We then train a small-scale reference model (e.g., 0.5B parameters)

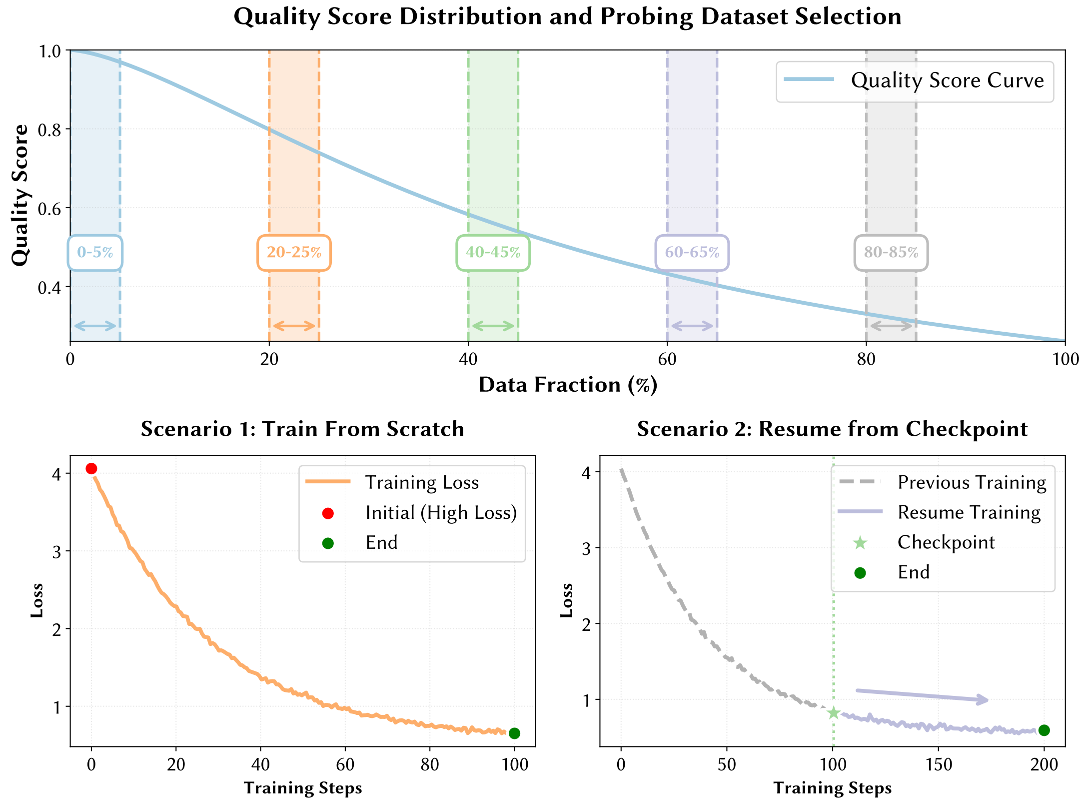


Figure 3: Illustration of Quantile Benchmarking Process.

on each of these probing datasets. Finally, we evaluate the resulting models on a set of target benchmarks to record their performances. We refer to this process of evaluating datasets across different quality quantiles as *quantile benchmarking*.

Given the computational cost of training multiple reference models on different probing datasets, we typically apply quantile benchmarking to dominant datasets, such as DCLM Baseline [43] and FineWeb-Edu [61] in the English domain, and FineWeb-Edu-Chinese-V2.1 [81] in the Chinese domain. Moreover, we measure the utility of each probing dataset in two scenarios: (1) training the reference model from scratch, and (2) continuing training from pretrained checkpoints. Evaluating both scenarios provides a more comprehensive understanding of the target dataset. Figure 3 illustrates the overall quantile benchmarking process, including quantile data selection and benchmarking experiments under both scenarios.

Results and Observations. Based on our quantile experiment results, we compare models trained on different dataset partitions across various benchmarks. These quantile-based comparisons provide deeper insights into the characteristics of target datasets and offer guidance for data selection and mixing strategies.

As an illustrative example, we present quantile experiments on both Fineweb-Edu and DCLM Baseline, offering a complementary perspective to previous analyses [67, 74]. The representative comparison sees Figure 4 and full comparison results can refer to Figures 9 and 10. Our investigation aims to deepen the understanding of these representative open-source datasets and identify their key differences and commonalities, which we summarize as follows:

(1) **Task-dependent dataset superiority.** Fineweb-Edu generally demonstrates superior performance on academic and encyclopedic benchmarks, including MMLU [31] and Common Sense QA (CSQA) [69], as well as reading comprehension tasks like BoolQ [12]. In contrast, DCLM Baseline exhibits slight advantages on situated commonsense reasoning, such as PIQA [10], Social IQa [64], HellaSwag [85], and WinoGrande [63]. This divergence suggests that FineWeb-Edu excels in tasks requiring more structural knowledge and formal semantics, while DCLM may benefit tasks relying on intuitive scenario-based reasoning. Representative comparisons are illustrated in Figure 4 where Fineweb-Edu induces better results on MMLU while DCLM Baseline outperforms on WinoGrande. More comprehensive results

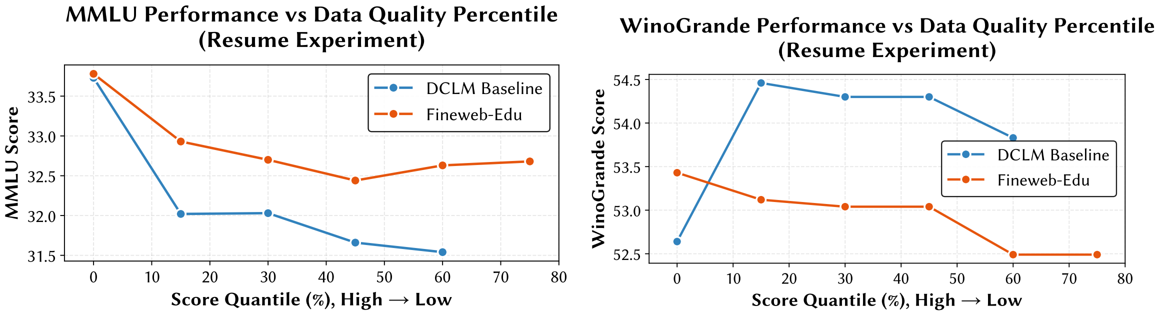


Figure 4: Representative results from quantile benchmarking experiments comparing Fineweb-Edu and DCLM Baseline across different quality quantiles.

are presented in Figure 10 and Figure 9, which respectively highlight academic knowledge and formal reasoning, and situated commonsense reasoning.

(2) **Substantial within-dataset heterogeneity.** Data quality varies considerably within individual datasets. For instance, in the continual training (resume) scenario, DCLM Baseline exhibits a 2% performance difference on MMLU between the top 0% and top 60% quantiles, while showing an even more pronounced 8% variation on ARC-Easy across the same quality range, as shown in Table 3. This substantial heterogeneity underscores the importance of quality-aware data selection and training strategies.

(3) **Consistency across training scenarios.** The relative superiority relationships between datasets remain largely consistent across both continual training (resume) and from-scratch (run) scenarios, as demonstrated in Figure 9 and Figure 10. However, we observe occasional deviations in specific quantile ranges, suggesting that training dynamics may influence relative dataset effectiveness.

(4) **Non-monotonic quality-performance relationships.** Benchmark performance does not necessarily increase monotonically with quality scores. As shown in Figure 9 and Figure 10, increasing quality scores, measured by the fineweb-edu classifier for Fineweb-Edu and the FastText score for DCLM Baseline, can paradoxically lead to decreased performance on HellaSwag and PIQA. This finding calls into question the universal applicability of quality metrics employed in current leading open-source datasets, and highlights the task-specific nature of data quality assessment.

In summary, our quantile experiments reveal that (i) datasets exhibit substantial internal heterogeneity, and (ii) the relative superiority of both datasets and their quality partitions is highly dependent on the target capability of interest. Quality assessment is inherently relative rather than absolute, precluding rigorous universal comparisons between open-source datasets. More details of implementation and discussions are presented in Section E.4.

These findings inform our data mixing and training strategies in the following ways:

(1) **Curriculum learning with selective repetition.** Beyond the conventional practice of filtering low-quality data, we propose strategically scheduling high-quality data partitions toward later training stages (curriculum learning) while applying higher repetition rates to these partitions compared to lower-quality data (selective multi-epoch training). This approach leverages within-dataset quality variation to enhance training efficiency. (Detailed in Section 4)

(2) **Benchmark-guided mixing ratios.** Given a representative benchmark aligned with a target capability, such as MMLU for knowledge-intensive tasks, quantile comparisons can guide inter-dataset mixing ratios. For example, as illustrated in the left panel of Figure 4, the entire Fineweb-Edu dataset exhibits performance roughly comparable to the top 30% partition of DCLM Baseline on MMLU, suggesting appropriate relative sampling rates for knowledge-focused pretraining. In practice, as shown in Table 7, in phase 2, we use the whole Fineweb-Edu dataset while using only the top 33.4% DCLM-Baseline dataset. In the latter phase, the relative ratio of DCLM-Baseline is further pulled down, shown in Tables 8 to 10.

We acknowledge that the current analysis remains primarily qualitative and coarse-grained. More fine-grained, quantitative frameworks for dataset comparison and mixing ratio optimization represent promising directions for future research.

Table 1: Performance Comparison: Curriculum Learning Strategies

Method	Retain	MMLU	ARC-c	ARC-e	CSQA	OBQA	PIQA	SIQA	Wino.	Avg.	Core
Uniform	100%	30.77	42.14	61.05	50.86	45.20	72.42	45.75	56.27	50.56	46.21
CMA	100%	31.68	**41.47**	**61.93**	**52.50**	**46.00**	71.71	45.39	57.22	**50.99**	**46.89**
Filter&Repeat	13.8%	**32.99**	35.79	61.75	46.03	42.00	71.71	44.37	56.35	48.87	44.14
Filter&Repeat	33.4%	32.44	41.14	**61.93**	51.11	43.80	72.09	45.34	**58.80**	50.83	46.65
Filter&Repeat	77.4%	31.68	38.46	60.70	**52.50**	45.00	**72.52**	**45.80**	57.22	50.49	45.83

3.2 Data Processing Framework

To address the challenges of data processing, our data processing framework is designed to satisfy three critical requirements:

1. **Reproducibility:** Given that the training dataset of Kaiyuan-2B is composed of various open-source datasets, the framework should be able to reconstruct the exact dataset from these original sources with a configuration file.
2. **Usability and Scalability:** The framework should support various operations like filtering, deduplication and mixing. Furthermore, this framework should scale to large clusters without additional engineer efforts.
3. **High Performance:** To handle hundreds of terabytes of data, the framework must be optimized to reduce computation overhead.

To meet these demands, we developed **Kaiyun-Spark**, a distributed data processing framework built on Spark [84]. Kaiyun-Spark adopts a tree-structured processing pipeline design. The leaf nodes represent the raw open-source datasets, while internal nodes represent processing operators like filters and samplers. The root node generates the final mixed training dataset. With this design, the entire processing pipeline, including dataset sources and operator parameters, can be defined with a YAML configuration file. This ensures strict reproducibility, enabling researchers to reconstruct the exact training corpus from raw datasets simply by applying the configuration.

As Kaiyun-Spark is built on Spark, it inherits the programming flexibility and scalability. We utilize the powerful Spark RDD API to develop complex data processing operators, and rely on the Spark Engine for distributed processing, resource management, and fault tolerance. This design allows Kaiyun-Spark to process over 100 TB of data across large-scale clusters with minimal engineering efforts.

Despite Spark's scalability, the overhead of JVM-based execution can become a bottleneck for compute-intensive tasks. To address this, we integrated the Chukonu [80] framework, utilizing its C++ interface to refactor certain performance-critical operators. By conducting computations with native C++, we accelerates the processing procedure. For instance, the optimized MinHash deduplication operator is approximately 2.5$\times$ faster than the Spark implementation.

4 Multi-Phase Multi-Domain Curriculum Training

Data quality heterogeneity within datasets, as revealed in Section 3, presents both opportunities and challenges for model training. High-quality samples can significantly enhance model capabilities more efficiently than average-quality data, yet they typically constitute only a small fraction of the overall dataset. To leverage this heterogeneity, we propose two principles: (1) progressive exposure, where higher-quality data appears in later training phases, and (2) strategic repetition, where high-quality partitions will be repeated more. In implementation, we design data curriculum at both phase and instance levels, and repeat data across different phases, thereby amplifying the impact of valuable training samples and improving overall data utilization efficiency. The multi-phase training practice is also adopted in other open-source model training pipelines [1, 36].

4.1 Multi-Phase Data Mixture

We structure the training process into five distinct phases with progressive data mixtures, as illustrated in Figure 5. This phased approach incorporates two perspectives of curriculum strategies: domain-level progression and quality-based selection and repetition.

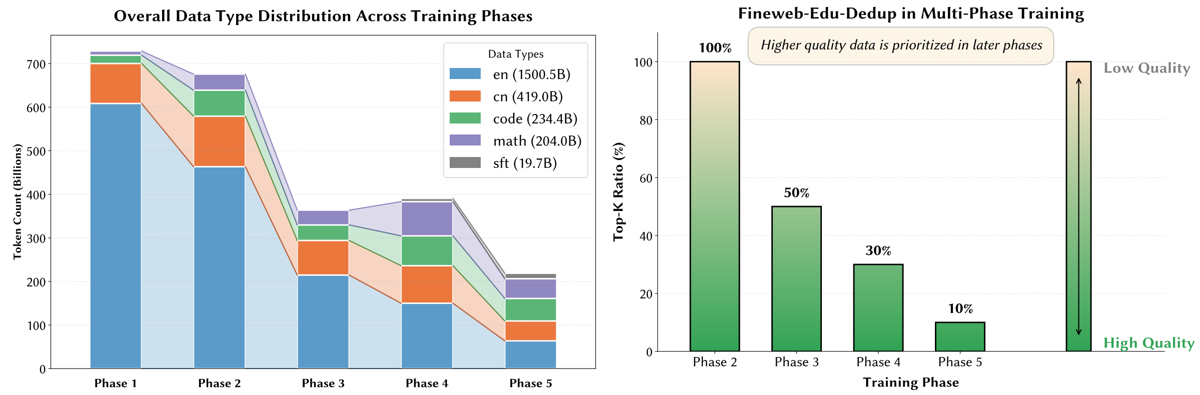


Figure 5: **Left:** phase-wise data mixture transitions. **Right:** phase-wise top-k ratio for Fineweb-Edu dataset. Latter phases keep more refined data samples.

First, we implement a domain-level curriculum by gradually increasing the proportion of Chinese, code, and mathematical datasets in later phases, while introducing supervised fine-tuning data in the final two phases. The phase-wise mixture transitions are visualized in Figure 5. To keep training stability, we maintain English content above 30% while limiting Chinese, code, and mathematical content each below 30%. The specific domain mixtures are detailed in Tables 6–10.

Second, we apply quality-based filtering within each domain during later phases, retaining only high-scoring partitions based on available quality metrics. Specifically, one dataset can occur in multiple phases rather than appear only once in the whole training process. However, in each phase, each data sample mostly occurs only once. Moreover, instead of repeating the whole dataset, we mostly keep only the high-quality partition and progressively decrease top-k retention ratios across phases, effectively increasing average data quality, for datasets with a quality metric. For example, as shown in Figure 5, we use the entire Fineweb-Edu dataset in phase two, then retain only 50%, 30%, and 10% of top-quality samples in subsequent phases. Consequently, the highest-quality 10% of samples repeat four times throughout training, while lower-quality samples appear only once during earlier phases. Higher-quality data samples are repeated more frequently.

This selective repetition serves two primary purposes. On the one hand, we experimentally find that mildly repeating a high-quality portion can attain better training efficiency than one-pass training. We validate this approach using a 1.5B Qwen2.5 model trained on 30B tokens from a DCLM Baseline shard. As shown in Table 1, retaining 33.4% of top-quality samples for three epochs outperforms one-pass training, demonstrating the efficacy of strategic repetition. Experimental details see Section E.5. On the other hand, repetition compensates for aggressive deduplication, as high-quality content naturally occurs more frequently in the internet and can also serve as an indicator of data quality. Prior research indicates that mild multi-epoch training (under four repetitions) preserves sample utility, with larger datasets tolerating more repetition [75, 52].

4.2 Multi-Domain Data Curriculum

Beyond phase-level adjustments, we construct instance-level curriculum learning within each phase. To fully take advantage of the data curriculum, we adopt the technique of Curriculum Model Average (CMA) [48], which adopts appropriate learning rate scheduling and model averaging in curriculum-based pretraining. As demonstrated in Table 1, CMA outperforms uniform sampling in our 1.5B model experiments. We discuss the small-scale reference experiment in Section E.5 in detail.

However, the pretraining dataset mostly consists of data samples from various source corpora and constructing a multi-dataset curriculum will present new challenges. Different datasets may employ distinct quality metrics or lack them entirely. To address this, we propose the three-step procedure outlined in Algorithm 1 and Figure 6:

1. **Within-Dataset Ranking**: Samples within each dataset are independently sorted using dataset-specific quality metrics in ascending order. For samples without quality labels, we can add a random number between 0 and 1, and then sort by these random scores.
2. **Rank Rescaling**: Dataset-specific ranks are normalized to a global scale using:

$$R_{\text{global}}(x_A) = r_A \times \frac{N_{\text{total}}}{N_A}$$

Algorithm 1 Multi-Dataset Curriculum Construction

Require: Datasets $D_1, D_2, \ldots, D_k$ with their specific quality metrics
Ensure: Multi-dataset curriculum dataset

```
N_total ← Σ_{i=1}^{k} |D_i|                                          ▷ Compute total sample count
for each dataset D_i do
    (Optionally) Add a random number for the dataset without a quality label
    Sort D_i by dataset-specific quality metric (ascending)          ▷ Within-dataset ranking
    Assign ordinal ranks r_i(x) ∈ [1, |D_i|] to each sample x ∈ D_i
    Compute rescaled ranks: R(x) ← r_i(x) × N_total / |D_i| for all x ∈ D_i
end for
U ← ⋃_{i=1}^{k} D_i                                                  ▷ Combine all datasets
Sort U by rescaled rank R(x) in ascending order                      ▷ Global interleaving
return sorted U
```

where r_A is the within-dataset rank, N_A is the dataset sample count, and N_{total} is the total sample count.

3. **Global Interleaving**: All samples are merged and sorted by their rescaled global ranks.

This algorithm ensures: (1) preservation of within-dataset quality ordering or shuffling the dataset without quality labels, (2) proportional interleaving across datasets according to mixture ratios, and (3) maintenance of stable dataset mixtures throughout training.

In practice, we implement this multi-dataset curriculum in the final three training phases to avoid low-quality samples being sorted together and fed to an immature model, which can result in instability. We set the final learning rate to 6×10^{-4} and average the last six checkpoints, as detailed in Section 4.3.

4.3 Pretraining Configuration

Model Architecture. Our 2B-parameter model architecture primarily refers to Qwen3-1.7B [76] with modifications for training stability. We untie word embeddings to reduce communication overhead, resulting in 1.4B non-embedding parameters and 0.6B embedding parameters. To ensure FP16 training stability, we incorporate QK-norm, sandwich norm, and soft capping (detailed in Section 2). Complete architectural specifications are provided in Table 11.

Training Hyperparameters. We train with a context length of 4096 and a batch size of 2048. We run small-scale experiments with a 1.5B model on 30B tokens at a batch size of 512, and then extrapolate roughly an optimal peak learning rate of 5×10^{-3} via square-root scaling [49]. We employ a Warmup-Stable-Decay schedule [35] and reduce the peak learning rate from 5×10^{-3} to 3×10^{-3} after the first phase to mitigate the instability caused by data distribution shift effects. The learning rate remains constant through phases 2–4, then decays to 6×10^{-4} in phase 5 to accommodate the multi-dataset curriculum [48]. We average the final eight checkpoints (saved every 3.36B tokens) to reduce variance from insufficient learning rate decay, with model averaging details provided in Section E.3.

Training Curve Analysis. Figure 7 presents learning rate, training loss, and validation loss trajectories. Two key observations emerge:

1. Training loss exhibits non-standard decay patterns due to three factors: (1) the learning rate reductions between phase 1 and phase 2 induce abrupt loss drop at the phase transition, (2) in later phases, we introduce increased low-perplexity code and mathematical content, which results in continually decreasing loss, and (3) the quality-based curricula import more high-quality data in latter steps within phases 3–5, which accelerates convergence of loss, followed by slight increases at phase transitions.
2. Validation loss on a DCLM Baseline subset shows similar phase-transition drops but anomalous increases during phases 3–4. These increases likely reflect domain misalignment between the validation set (primarily English text) and training data (increasing code and mathematical content). Each phase ends with accelerated validation loss decay (benefiting from high-quality data) followed by sharp increases (probably from distribution shifts by curriculum).

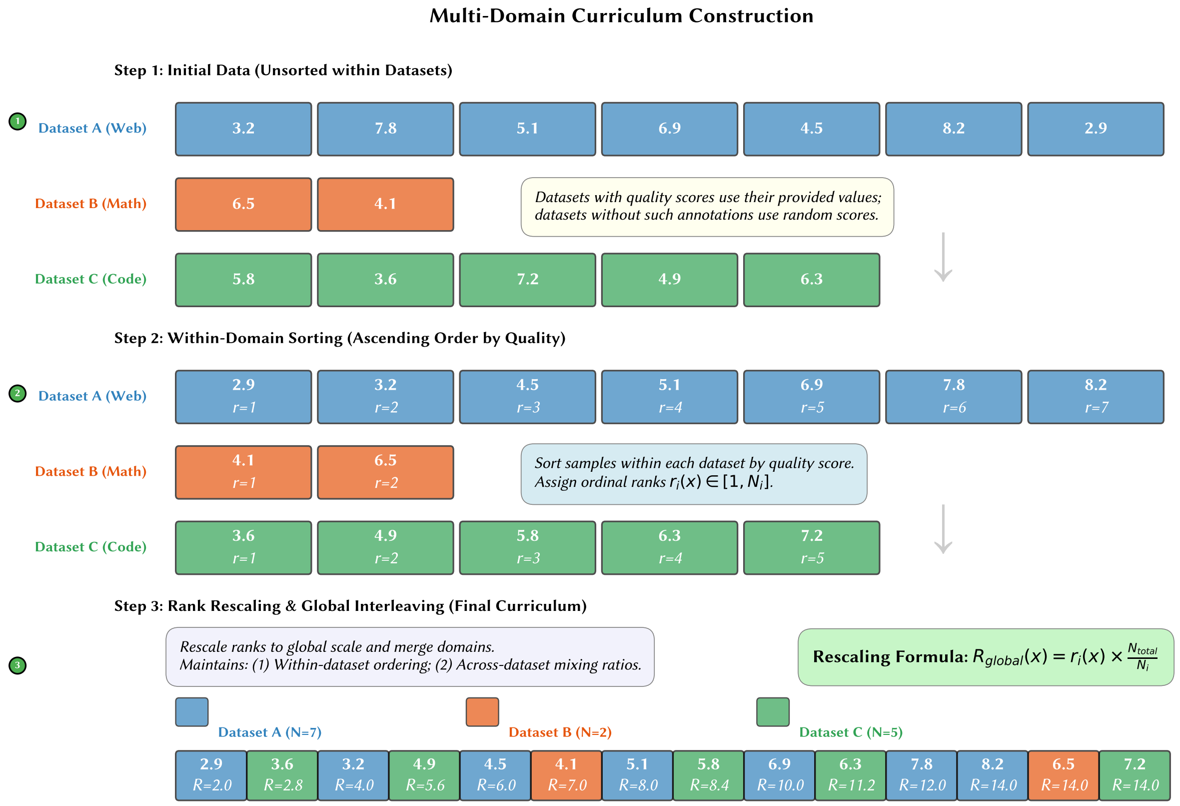


Figure 6: Multi-Dataset Curriculum Construction Process

These patterns suggest that more diverse validation sets would better track training progress. Future work should consider more gradual domain transitions and increased phase counts for improved training stability. Using a domain data schedule that shifts dataset domains smoothly, like an LR schedule, can be a promising future practice.

5 Evaluation

5.1 Evaluation Setup

5.1.1 Baseline Models

We evaluate Kaiyuan-2B against a comprehensive set of state-of-the-art baseline models with comparable parameter counts. These baselines are categorized into two distinct groups: **open-weight models**, where model weights are public but training data or details may remain proprietary, and **fully-open models**, where the architecture, weights, training code, and datasets are all publicly released.[1]

Open-weight models.

- *Qwen2-1.5B* [77]: A 1.5B-parameter decoder-only transformer trained on large-scale multilingual and code data. It delivers robust performance in general language understanding, coding, and reasoning while facilitating efficient deployment.

[1]All evaluated models are base (pretrained) checkpoints without instruction tuning. To maintain consistency, we standardize naming conventions by omitting suffixes (e.g., using simplified names for Qwen3) to denote base models throughout this paper.

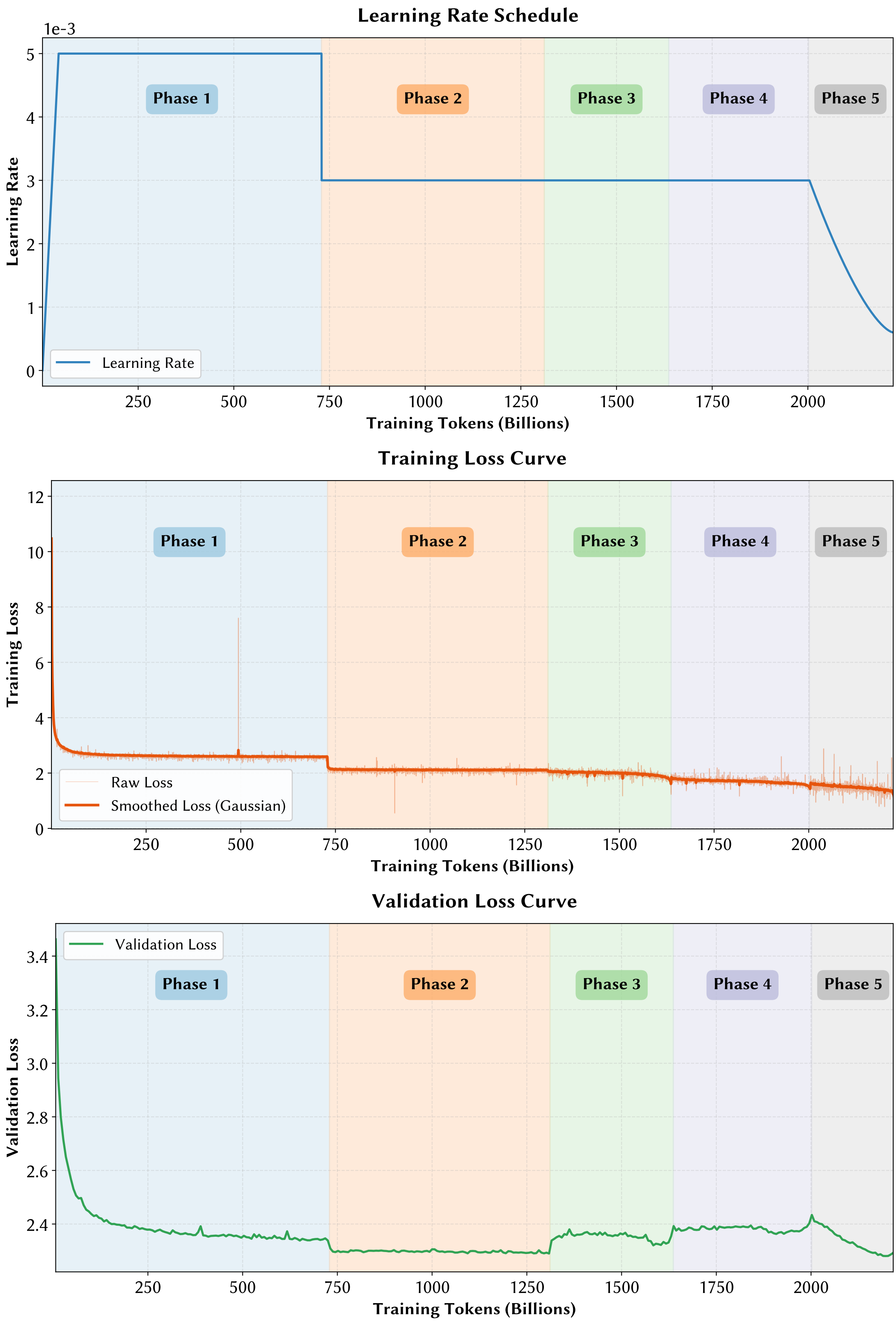


Figure 7: Learning Rate Schedule, Training Loss, and Validation Loss Curves

- *Qwen2.5 series* [78]: We select Qwen2.5-1.5B and Qwen2.5-3B, dense foundation models that refine the Qwen2.5 architecture. These models feature an improved tokenizer and offer enhanced capabilities in knowledge, coding, and mathematics within a compact form factor suitable for edge applications.
- *Qwen3 series* [76]: We include Qwen3-0.6B, Qwen3-1.7B, and Qwen3-4B. These are small-scale base models that support long contexts and "thinking" modes, providing competitive abilities in general tasks, mathematics, and coding.
- *Gemma2-2B* [62]: The smallest member of Google's Gemma 2 family, this model is distilled from larger counterparts. It was trained on 2 trillion tokens from diverse sources, including web documents, code, and scientific articles.
- *Llama3.2 series* [50]: We utilize Llama-3.2-1B and Llama-3.2-3B, multilingual text-only models distilled and pruned from larger Llama variants. They support extended context windows (128k) and tool-calling, targeting privacy-preserving on-device inference.

Fully-open models.

- *SmolLM2-1.7B* [1]: Developed by Hugging Face, this model utilizes the Llama 2 architecture with a GPT-2 tokenizer (vocabulary size 49,152). It was trained on 256 H100 GPUs.
- *SmolLM3-3B* [7]: A compact, fully-open model trained on 11T tokens using data-centric recipes. It features a 128k context window utilizing NoPE and YaRN, offering state-of-the-art performance for its size class with multilingual support.
- *OLMo2-1B* [57]: The smallest model in the OLMo2 family (specifically OLMo2-0425-1B), trained on the OLMo-mix corpus. Its full release of code, checkpoints, logs, and training details enables rigorous scientific inquiry into compute-efficient training at the 1B-parameter scale.
- *YuLan-Mini* [79]: A 2.4B-parameter model pre-trained on approximately 1.08T tokens. By combining curated data pipelines with robust optimization and annealing strategies, it achieves top-tier performance among similarly sized models, particularly in mathematics and coding.

5.1.2 Benchmarks

Our evaluation encompasses four primary domains: mathematics, coding, Chinese language processing, and general reasoning & knowledge. We selected representative benchmarks for each domain as follows:

- *Math*: We utilize GSM8K [14] and MATH [32]. Together, these datasets cover the spectrum from grade-school arithmetic to advanced competition-style problems, providing a comprehensive assessment of symbolic and multi-step reasoning.
- *Coding*: We adopt MBPP [5] and HumanEval [11] to evaluate code generation via unit testing. This approach directly measures the model's ability to synthesize executable and logically coherent programs. specifically, we use the sanitized subset of MBPP, which refines problem descriptions and test cases to minimize ambiguity.
- *Chinese*: To assess knowledge and reasoning within a Chinese linguistic context, we employ CMMLU [41] and C-Eval [37], widely adopted benchmarks spanning diverse academic and professional subjects.
- *Reasoning & Knowledge*: For general English-language knowledge and reasoning, we include a suite of eight datasets: MMLU [31], HellaSwag [85], Common Sense QA (CSQA) [69], BoolQ [12], PIQA [10], SocialIQA [64], WinoGrande [63], and ARC [13]. These benchmarks cover a broad range of expert knowledge, reading comprehension, and commonsense reasoning scenarios.

5.1.3 Implementation Details

We conduct our evaluation using the OpenCompass framework [18], a comprehensive platform for large model evaluation. For mathematics and coding benchmarks, which typically require open-ended generation, we evaluate models in *generation mode*. Conversely, for benchmarks in other domains, we employ *perplexity-based (PPL) evaluation*. Following the OLMES protocol [25], PPL tasks are assessed under both multiple-choice formulation (MCF) and completion formulation (CF), with the superior score reported as the final result.

5.2 Evaluation Results

The performance of Kaiyuan-2B and baseline models is summarized in Table 2 and Table 3, with a comprehensive comparison provided in Table 17.

Core Capabilities: Math, Code, and Chinese. Table 2 highlights the model's core capabilities, defined here as proficiency in mathematics, coding, and Chinese language tasks.[2] Kaiyuan-2B achieves an average score of `46.05` across these seven benchmarks. It outperforms fully-open models of similar scale, such as SmolLM2-1.7B and OLMo-2-0425-1B, and remains competitive with larger models like YuLan-Mini-2.4B and SmolLM3-3B despite a smaller parameter count. Specifically, on Chinese benchmarks (C-Eval and CMMLU), Kaiyuan-2B scores `46.30` and `49.25`, respectively—markedly higher than SmolLM2-1.7B and OLMo-2-0425-1B, and approaching the performance of the larger SmolLM3-3B. In mathematics, Kaiyuan-2B achieves `51.33` on GSM8K, substantially surpassing SmolLM2-1.7B, and scores `30.34` on MATH, outperforming YuLan-Mini-2.4B (`27.12`). Similarly, in code generation, our model reaches `42.68` on HumanEval, exceeding both SmolLM3-3B (`39.63`) and Qwen2.5-3B (`42.10`). These results demonstrate that Kaiyuan-2B offers a superior trade-off between accuracy and model size in critical domains.

Table 2: Core Capabilities: Language (Chinese & English), Math, and Code. Scores marked with * are cited from their official report or paper.

Model Name	**Params**	**Chinese**		**Math**		**Code**		**Avg**
		C-Eval 5 shot	CMMLU 5 shot	GSM8K 4 shot	MATH 4 shot	sanitized-MBPP 3 shot	HumanEval 3 shot	
Open-Weight SOTA								
Qwen2-1.5B	1.5B	71.29	70.62	58.50*	21.70*	50.58	31.10*	50.63
Qwen2.5-1.5B	1.5B	68.63	68.01	68.50*	35.00*	58.37	37.20*	55.95
Qwen2.5-3B	3B	74.65	73.92	79.10*	42.60*	66.54	42.10*	63.15
Qwen3-0.6B	0.6B	57.03	52.36	59.59*	32.44*	51.75	29.88	47.18
Qwen3-1.7B	1.7B	66.70	66.55	75.44*	43.5*	64.20	52.44	61.47
Qwen3-4B	4B	78.5	77.01	87.79*	54.10*	74.32	62.20	72.32
gemma2-2B	2B	41.35	39.63	23.90*	15.00*	38.91	17.70*	29.42
llama-3.2-1B	1B	29.82	31.03	44.40*	30.60*	34.63	18.90	31.56
llama-3.2-3B	3B	45.67	44.33	77.70*	48.00*	49.42	29.88	49.17
Fully-Open SOTA								
SmolLM2-1.7B	1.7B	35.06	34.03	31.10*	11.60*	49.42	22.60*	30.64
OLMo-2-0425-1B	1B	30.53	28.62	68.30*	20.70*	15.56	6.71	28.40
YuLan-Mini-2.4B	2.4B	52.32	48.14	66.65*	27.12	62.26	61.60*	53.02
SmolLM3-3B	3B	50.84	49.35	67.63*	46.10*	62.26	39.63	52.64
Ours								
PCMind-2.1-Kaiyuan-2B	2B	46.30	49.25	51.33	30.34	56.42	42.68	46.05

Reasoning and Knowledge. Table 3 presents performance on nine reasoning and knowledge benchmarks. Kaiyuan-2B achieves an average score of `67.74`, placing it firmly within the competitive range for its size class. Within the fully-open category, our model surpasses SmolLM2-1.7B (+1.69 average) and OLMo-2-0425-1B (+5.68 average), while effectively matching the larger YuLan-Mini-2.4B (`67.50`). Although the larger SmolLM3-3B attains a higher average (`72.60`), Kaiyuan-2B significantly narrows the gap to the state-of-the-art for fully-open models at this scale. When compared to open-weight models, Kaiyuan-2B is only slightly behind Gemma2-2B (`67.74` vs. `69.16`). Larger open-weight models like Qwen3-4B maintain a substantial lead (`81.84`), which is expected given their significantly larger scale and training resources.

Discussion on Size and Performance Trade-offs. The overall trade-off between model size and average benchmark performance is visualized in Figure 1. The figure reveals that Kaiyuan-2B lies beyond the current fully-open frontier: at comparable parameter counts, it clearly outperforms earlier fully-open models (e.g., OLMo-2-1B, SmolLM2-1.7B) and approaches the performance of the larger YuLan-Mini-2.4B. Moreover, if adhering to the convention of comparing non-embedding parameters to get rid of the vocabulary effect, our Kaiyuan-2B can exhibit even more prominent advantages, as shown in Figure 8. We compare different models according to non-embedding parameters in Section A.

[2] For generation tasks (math and code), we report official results for baseline models where available, as exact reproduction can be challenging.

Table 3: Reasoning and Knowledge Capabilities

Model Name	Params	Reasoning & Knowledge									Avg
		MMLU 5 shot	ARC-C 5 shot	ARC-E 5 shot	BoolQ 5 shot	CSQA 5 shot	HSwag 5 shot	PIQA 5 shot	SocIQ 5 shot	Wino 5 shot	
Open-Weight SOTA											
Qwen2-1.5B	1.5B	56.36	70.17	83.60	71.90	70.52	60.77	75.73	63.46	59.83	68.04
Qwen2.5-1.5B	1.5B	61.56	79.32	90.48	76.39	75.10	64.18	76.17	64.94	59.67	71.98
Qwen2.5-3B	3B	66.86	86.44	92.59	83.88	76.09	73.85	81.45	69.40	63.69	77.14
Qwen3-0.6B	0.6B	55.09	68.14	84.48	69.05	61.18	48.51	69.97	61.51	55.64	63.73
Qwen3-1.7B	1.7B	65.35	80.34	91.89	79.82	74.61	60.76	77.20	68.58	59.27	73.09
Qwen3-4B	4B	75.78	89.83	97.53	86.09	81.9	79.46	84.98	75.59	65.43	81.84
gemma2-2B	2B	55.20	66.44	82.54	72.42	69.45	66.20	78.89	65.92	65.35	69.16
llama-3.2-1B	1B	37.74	36.95	70.55	67.43	62.82	60.20	74.92	50.61	58.17	57.71
llama-3.2-3B	3B	57.87	72.20	83.95	76.73	70.35	71.06	79.05	64.33	64.09	71.07
Fully-Open SOTA											
SmolLM2-1.7B	1.7B	51.99	59.66	82.72	69.85	67.16	65.30	78.51	60.18	59.12	66.05
OLMo-2-0425-1B	1B	44.25	47.46	76.72	70.55	65.60	61.61	76.44	55.53	60.38	62.06
YuLan-Mini-2.4B	2.4B	51.76	64.75	82.54	78.59	66.18	61.20	77.31	63.25	61.88	67.50
SmolLM3-3B	3B	63.04	77.29	88.54	76.12	70.52	69.20	79.05	65.25	64.40	72.60
Ours											
PCMind-2.1-Kaiyuan-2B	2B	53.90	66.10	82.89	78.53	67.40	58.13	74.37	62.59	65.75	67.74

Furthermore, when compared to open-weight baselines of similar size, Kaiyuan-2B demonstrates superior architectural efficiency. For instance, while Gemma2-2B uses tied embeddings, Kaiyuan-2B utilizes non-tied embeddings. Consequently, the non-embedding parameters in our model count only 1.4B compared to 2B in Gemma2-2B, despite total parameter counts of 2B and 2.6B, respectively. Moreover, Kaiyuan-2B is trained on 2.2T tokens, comparable to Gemma2-2B's reported 2T tokens. As shown in Tables 2, 3 and 17, Kaiyuan-2B leverages this efficiency to achieve stronger performance on core capabilities (Chinese, Math, Code) and competitive reasoning scores with fewer total parameters. Although a gap remains compared to the Qwen series, likely due to their massive training data scale (e.g., 36T tokens), Kaiyuan-2B occupies a favorable position in the size-performance landscape, offering a strong, fully-open alternative for resource-constrained environments.

6 Conclusion

The Kaiyuan-2B project successfully demonstrates a systematic and resource-efficient approach to fully open-source LLM pretraining, providing concrete answers to the challenges of data heterogeneity and computational scarcity. Our core contributions include Quantile Data Benchmarking, Strategic Manual Repetition, and Multi-Domain Curriculum Training. Together, they represent a practical framework for the academic community to select and utilize public data effectively. By releasing the model checkpoint, the open-source data preprocessing framework, and the final pretraining dataset, we provide a complete, transparent recipe for high-quality LLM pretraining. We believe Kaiyuan-2B is a valuable contribution that will facilitate further exploration and innovation in the open-source LLM ecosystem, pushing the frontier of what is achievable under limited resources.

References

[1] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlícek, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. 2025. SmolLM2: When Smol Goes Big - Data-Centric Training of a Small Language Model. *CoRR* abs/2502.02737 (2025). arXiv:2502.02737 doi:10.48550/ARXIV.2502.02737

[2] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. 2025. SmolLM2: When Smol Goes Big – Data-Centric Training of a Small Language Model. arXiv:2502.02737 [cs.CL] https://arxiv.org/abs/2502.02737

[3] arXiv info. 2025. License and copyright - arXiv info — info.arxiv.org. https://info.arxiv.org/help/license/index.html. [Accessed 03-12-2025].

[4] arXiv info. 2025. Terms of Use for arXiv APIs - arXiv info — info.arxiv.org. https://info.arxiv.org/help/api/tou.html. [Accessed 03-12-2025].

[5] Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. 2021. Program Synthesis with Large Language Models. *CoRR* abs/2108.07732 (2021). arXiv:2108.07732 https://arxiv.org/abs/2108.07732

[6] Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2023. Llemma: An Open Language Model For Mathematics. arXiv:2310.10631 [cs.CL]

[7] Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patiño, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, Kashif Rasul, Nathan Habib, Clémentine Fourrier, Hynek Kydlicek, Guilherme Penedo, Hugo Larcher, Mathieu Morlon, Vaibhav Srivastav, Joshua Lochner, Xuan-Son Nguyen, Colin Raffel, Leandro von Werra, and Thomas Wolf. 2025. SmolLM3: smol, multilingual, long-context reasoner. https://huggingface.co/blog/smollm3.

[8] Irwan Bello, Hieu Pham, Quoc V. Le, Mohammad Norouzi, and Samy Bengio. 2017. Neural Combinatorial Optimization with Reinforcement Learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=Bk9mxlSFx

[9] Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. 2024. *SmolLM-Corpus*. https://huggingface.co/datasets/HuggingFaceTB/smollm-corpus

[10] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about Physical Commonsense in Natural Language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020.* AAAI Press, 7432–7439. doi:10.1609/AAAI.V34I05.6239

[11] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. *CoRR* abs/2107.03374 (2021). arXiv:2107.03374 https://arxiv.org/abs/2107.03374

[12] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 2924–2936. doi:10.18653/v1/N19-1300

[13] 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. *CoRR* abs/1803.05457 (2018). arXiv:1803.05457 http://arxiv.org/abs/1803.05457

[14] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. *CoRR* abs/2110.14168 (2021). arXiv:2110.14168 https://arxiv.org/abs/2110.14168

[15] Common Crawl. 2024. Common Crawl - Terms of Use — commoncrawl.org. https://commoncrawl.org/terms-of-use. [Accessed 03-12-2025].

[16] OpenCSG Community. 2024. OpenCSG Model Community License. https://huggingface.co/datasets/opencsg/chinese-fineweb-edu/blob/main/opencsg%E6%A8%A1%E5%9E%8B%E7%A4%BE%E5%8C%BA%E8%AE%B8%E5%8F%AF%E5%8D%8F%E8%AE%AE.pdf. [Accessed 03-12-2025].

[17] Together Computer. 2023. *RedPajama: An Open Source Recipe to Reproduce LLaMA training dataset.* https://github.com/togethercomputer/RedPajama-Data

[18] OpenCompass Contributors. 2023. OpenCompass: A Universal Evaluation Platform for Foundation Models. https://github.com/open-compass/opencompass.

[19] Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (Sydney, NSW, Australia) *(ICML'17)*. JMLR.org, 933–941.

[20] DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948 [cs.CL] https://arxiv.org/abs/2501.12948

[21] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. 2024. DeepSeek-V3 Technical Report. *CoRR* abs/2412.19437 (2024). arXiv:2412.19437 doi:10.48550/ARXIV.2412.19437

[22] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. 2021. CogView: Mastering Text-to-Image Generation via Transformers. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 19822–19835. https://proceedings.neurips.cc/paper/2021/hash/a4d92e2cd541fca87e4620aba658316d-Abstract.html

[23] Kazuki Fujii, Yukito Tajima, Sakae Mizuki, Hinari Shimada, Taihei Shiotani, Koshiro Saito, Masanari Ohi, Masaki Kawamura, Taishi Nakamura, Takumi Okamoto, Shigeki Ishida, Kakeru Hattori, Youmi Ma, Hiroya Takamura, Rio Yokota, and Naoaki Okazaki. 2025. Rewriting Pre-Training Data Boosts LLM Performance in Math and Code. arXiv:2505.02881 [cs.LG] https://arxiv.org/abs/2505.02881

[24] Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. 2024. OLMo: Accelerating the Science of Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 15789–15809. doi:10.18653/v1/2024.acl-long.841

[25] Yuling Gu, Oyvind Tafjord, Bailey Kuehl, Dany Haddad, Jesse Dodge, and Hannaneh Hajishirzi. 2025. OLMES: A Standard for Language Model Evaluations. In *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, Luis Chiruzzo, Alan Ritter, and Lu Wang (Eds.). Association for Computational Linguistics, 5005–5033. doi:10.18653/V1/2025.FINDINGS-NAACL.282

[26] Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro von Werra, and Martin Jaggi. 2024. Scaling Laws and Compute-Optimal Training Beyond Fixed Training Durations. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (Eds.). http://papers.nips.cc/paper%5Ffiles/paper/2024/hash/8b970e15a89bf5d12542810df8eae8fc-Abstract-Conference.html

[27] Conghui He, Zhenjiang Jin, Chao Xu, Jiantao Qiu, Bin Wang, Wei Li, Hang Yan, Jiaqi Wang, and Dahua Lin. 2023. WanJuan: A Comprehensive Multimodal Dataset for Advancing English and Chinese Large Models. arXiv:2308.10755 [cs.CL]

[28] Conghui He, Wei Li, Zhenjiang Jin, Chao Xu, Bin Wang, and Dahua Lin. 2024. OpenDataLab: Empowering General Artificial Intelligence with Open Datasets. arXiv:2407.13773 [cs.DL] https://arxiv.org/abs/2407.13773

[29] David Heineman, Valentin Hofmann, Ian Magnusson, Yuling Gu, Noah A. Smith, Hannaneh Hajishirzi, Kyle Lo, and Jesse Dodge. 2025. Signal and Noise: A Framework for Reducing Uncertainty in Language Model Evaluation. *CoRR* abs/2508.13144 (2025). arXiv:2508.13144 doi:10.48550/ARXIV.2508.13144

[30] Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021. Aligning AI With Shared Human Values. *Proceedings of the International Conference on Learning Representations (ICLR)* (2021).

[31] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021.* OpenReview.net. https://openreview.net/forum?id=d7KBjmI3GmQ

[32] 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. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, Joaquin Vanschoren and Sai-Kit Yeung (Eds.). https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html

[33] Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. 2020. Query-Key Normalization for Transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020 (Findings of ACL, Vol. EMNLP 2020)*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 4246–4253. doi:10.18653/V1/2020.FINDINGS-EMNLP.379

[34] Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. 2021. Scaling Laws for Transfer. *CoRR* abs/2102.01293 (2021). arXiv:2102.01293 https://arxiv.org/abs/2102.01293

[35] Shengding Hu, Yuge Tu, Xu Han, Ganqu Cui, Chaoqun He, Weilin Zhao, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Xinrong Zhang, Zhen Leng Thai, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie

Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, dahai li, Zhiyuan Liu, and Maosong Sun. 2024. MiniCPM: Unveiling the Potential of Small Language Models with Scalable Training Strategies. In *First Conference on Language Modeling*. `https://openreview.net/forum?id=3X2L2TFr0f`

[36] Yiwen Hu, Huatong Song, Jia Deng, Jiapeng Wang, Jie Chen, Kun Zhou, Yutao Zhu, Jinhao Jiang, Zican Dong, Wayne Xin Zhao, et al. 2024. YuLan-Mini: An Open Data-efficient Language Model. *arXiv preprint arXiv:2412.17743* (2024).

[37] 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. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.). `http://papers.nips.cc/paper%5Ffiles/paper/2023/hash/c6ec1844bec96d6d32ae95ae694e23d8-Abstract-Datasets%5Fand%5FBenchmarks.html`

[38] Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro von Werra, and Harm de Vries. 2022. The Stack: 3 TB of permissively licensed source code. *Preprint* (2022).

[39] Hynek Kydlíček, Guilherme Penedo, and Leandro von Werra. 2025. FinePDFs. `https://huggingface.co/datasets/HuggingFaceFW/finepdfs`.

[40] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. 2024. Tülu 3: Pushing Frontiers in Open Language Model Post-Training. (2024).

[41] Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024. CMMLU: Measuring massive multitask language understanding in Chinese. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, 11260–11285. doi:10.18653/V1/2024.FINDINGS-ACL.671

[42] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. 2024. DataComp-LM: In search of the next generation of training sets for language models. arXiv:2406.11794 [id='cs.LG' full_name='Machine Learning' is_active=True alt_name=None in_archive='cs' is_general=False description='Papers on all aspects of machine learning research (supervised, unsupervised, reinforcement learning, bandit problems, and so on) including also robustness, explanation, fairness, and methodology. cs.LG is also an appropriate primary category for applications of machine learning methods.']

[43] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Kumar Guha, Sedrick Scott Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee F. Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah M. Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Raghavi Chandu, Thao Nguyen, Igor Vasiljevic, Sham M. Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alex Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. 2024. DataComp-LM: In search of the next generation of training sets for language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M.

Tomczak, and Cheng Zhang (Eds.). http://papers.nips.cc/paper%5Ffiles/paper/2024/hash/19e4ea30dded58259665db375885e412-Abstract-Datasets%5Fand%5FBenchmarks%5FTrack.html

[44] Yunshui Li, Yiyuan Ma, Shen Yan, Chaoyi Zhang, Jing Liu, Jianqiao Lu, Ziwen Xu, Mengzhao Chen, Minrui Wang, Shiyi Zhan, Jin Ma, Xunhao Lai, Deyi Liu, Yao Luo, Xingyan Bin, Hongbin Ren, Mingji Han, Wenhao Hao, Bairen Yi, LingJun Liu, Bole Ma, Xiaoying Jia, Xun Zhou, Siyuan Qiao, Liang Xiang, and Yonghui Wu. 2025. Model Merging in Pre-training of Large Language Models. *CoRR* abs/2505.12082 (2025). arXiv:2505.12082 doi:10.48550/ARXIV.2505.12082

[45] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The Flan Collection: Designing Data and Methods for Effective Instruction Tuning. arXiv:2301.13688 [cs.AI]

[46] Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, Tianyang Liu, Max Tian, Denis Kocetkov, Arthur Zucker, Younes Belkada, Zijian Wang, Qian Liu, Dmitry Abulkhanov, Indraneil Paul, Zhuang Li, Wen-Ding Li, Megan Risdal, Jia Li, Jian Zhu, Terry Yue Zhuo, Evgenii Zheltonozhskii, Nii Osae Osae Dade, Wenhao Yu, Lucas Krauß, Naman Jain, Yixuan Su, Xuanli He, Manan Dey, Edoardo Abati, Yekun Chai, Niklas Muennighoff, Xiangru Tang, Muhtasham Oblokulov, Christopher Akiki, Marc Marone, Chenghao Mou, Mayank Mishra, Alex Gu, Binyuan Hui, Tri Dao, Armel Zebaze, Olivier Dehaene, Nicolas Patry, Canwen Xu, Julian J. McAuley, Han Hu, Torsten Scholak, Sébastien Paquet, Jennifer Robinson, Carolyn Jane Anderson, Nicolas Chapados, and et al. 2024. StarCoder 2 and The Stack v2: The Next Generation. *CoRR* abs/2402.19173 (2024). https://doi.org/10.48550/arXiv.2402.19173

[47] Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, Tianyang Liu, Max Tian, Denis Kocetkov, Arthur Zucker, Younes Belkada, Zijian Wang, Qian Liu, Dmitry Abulkhanov, Indraneil Paul, Zhuang Li, Wen-Ding Li, Megan Risdal, Jia Li, Jian Zhu, Terry Yue Zhuo, Evgenii Zheltonozhskii, Nii Osae Osae Dade, Wenhao Yu, Lucas Krauß, Naman Jain, Yixuan Su, Xuanli He, Manan Dey, Edoardo Abati, Yekun Chai, Niklas Muennighoff, Xiangru Tang, Muhtasham Oblokulov, Christopher Akiki, Marc Marone, Chenghao Mou, Mayank Mishra, Alex Gu, Binyuan Hui, Tri Dao, Armel Zebaze, Olivier Dehaene, Nicolas Patry, Canwen Xu, Julian McAuley, Han Hu, Torsten Scholak, Sebastien Paquet, Jennifer Robinson, Carolyn Jane Anderson, Nicolas Chapados, Mostofa Patwary, Nima Tajbakhsh, Yacine Jernite, Carlos Mu noz Ferrandis, Lingming Zhang, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2024. StarCoder 2 and The Stack v2: The Next Generation. arXiv:2402.19173 [cs.SE]

[48] Kairong Luo, Zhenbo Sun, Haodong Wen, Xinyu Shi, Jiarui Cui, Chenyi Dang, Kaifeng Lyu, and Wenguang Chen. 2025. How Learning Rate Decay Wastes Your Best Data in Curriculum-Based LLM Pretraining. arXiv:2511.18903 [cs.LG] https://arxiv.org/abs/2511.18903

[49] Sadhika Malladi, Kaifeng Lyu, Abhishek Panigrahi, and Sanjeev Arora. 2022. On the SDEs and Scaling Rules for Adaptive Gradient Algorithms. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.). http://papers.nips.cc/paper_files/paper/2022/hash/32ac710102f0620d0f28d5d05a44fe08-Abstract-Conference.html

[50] Meta AI. 2024. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models. https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/

[51] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, Brussels, Belgium, 2381–2391. doi:10.18653/v1/D18-1260

[52] Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A. Raffel. 2023. Scaling Data-Constrained Language Models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.).

[53] Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Evan Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, and et al. 2025. OLMoE: Open Mixture-of-Experts Language Models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net. https://openreview.net/forum?id=xXTkbTBmqq

[54] Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive Learning from Complex Explanation Traces of GPT-4. arXiv:2306.02707 [cs.CL]

[55] NVIDIA. 2025. NVIDIA Data Agreement for Model Training. https://huggingface.co/datasets/nvidia/Nemotron-Pretraining-Dataset-sample/blob/main/LICENSE.md. [Accessed 03-12-2025].

[56] Beijing Academy of Artificial Intelligence. 2023. Chinese Corpus Internet Usage Agreement. https://data.baai.ac.cn/resources/agreement/cci_usage_aggrement.pdf. [Accessed 03-12-2025].

[57] Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. 2 OLMo 2 Furious. *CoRR* abs/2501.00656 (2025). arXiv:2501.00656 doi:10.48550/ARXIV.2501.00656

[58] OpenAI. 2023. GPT-4 Technical Report. *CoRR* abs/2303.08774 (2023). arXiv:2303.08774 doi:10.48550/ARXIV.2303.08774

[59] Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. 2023. OpenWebMath: An Open Dataset of High-Quality Mathematical Web Text. arXiv:2310.06786 [cs.AI]

[60] Guilherme Penedo. 2025. *FineWiki*. https://huggingface.co/datasets/HuggingFaceFW/finewiki Source: Wikimedia Enterprise Snapshot API (https://api.enterprise.wikimedia.com/v2/snapshots). Text licensed under CC BY-SA 4.0 with attribution to Wikipedia contributors..

[61] Guilherme Penedo, Hynek Kydlícek, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin A. Raffel, Leandro von Werra, and Thomas Wolf. 2024. The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (Eds.). http://papers.nips.cc/paper%5Ffiles/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets%5Fand%5FBenchmarks%5FTrack.html

[62] Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozinska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucinska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjösund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, and Lilly McNealus. 2024. Gemma 2: Improving Open Language Models at a Practical Size. *CoRR* abs/2408.00118 (2024). arXiv:2408.00118 doi:10.48550/ARXIV.2408.00118

[63] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020.* AAAI Press, 8732–8740. doi:10.1609/AAAI.V34I05.6399

[64] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense Reasoning about Social Interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 4463–4473. doi:10.18653/v1/D19-1454

[65] Xiaofeng Shi, Lulu Zhao, Hua Zhou, and Donglin Hao. 2024. IndustryCorpus2. doi:10.57967/hf/3488

[66] Skywork-AI. 2023. Skywork Community License. https://huggingface.co/datasets/Skywork/SkyPile-150B/blob/main/Skywork%20Community%20License.pdf. [Accessed 03-12-2025].

[67] Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2025. Nemotron-CC: Transforming Common Crawl into a Refined Long-Horizon Pretraining Dataset. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, 2459–2475. https://aclanthology.org/2025.acl-long.123/

[68] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomput.* 568, C (Feb. 2024), 12 pages. doi:10.1016/j.neucom.2023.127063

[69] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 4149–4158. doi:10.18653/v1/N19-1421

[70] Gemini Team. 2025. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *CoRR* abs/2507.06261 (2025). arXiv:2507.06261 doi:10.48550/ARXIV.2507.06261

[71] Changxin Tian, Jiapeng Wang, Qian Zhao, Kunlong Chen, Jia Liu, Ziqi Liu, Jiaxin Mao, Wayne Xin Zhao, Zhiqiang Zhang, and Jun Zhou. 2025. WSM: Decay-Free Learning Rate Schedule via Checkpoint Merging for LLM Pre-training. *CoRR* abs/2507.17634 (2025). arXiv:2507.17634 doi:10.48550/ARXIV.2507.17634

[72] 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:2410.18505 [cs.CL] https://arxiv.org/abs/2410.18505

[73] Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, Haihua Yang, Biye Li, Cheng Cheng, Weiwei Lü, Rui Hu, Chenxia Li, Liu Yang, Xilin Luo, Xuejie Wu, Lunan Liu, Wenjun Cheng, Peng Cheng, Jianhao Zhang, Xiaoyu Zhang, Lei Lin, Xiaokun Wang, Yutuan Ma, Chuanhai Dong, Yanqi Sun, Yifu Chen, Yongyi Peng, Xiaojuan Liang, Shuicheng Yan, Han Fang, and Yahui Zhou. 2023. Skywork: A More Open Bilingual Foundation Model. arXiv:2310.19341 [cs.CL]

[74] Alexander Wettig, Kyle Lo, Sewon Min, Hannaneh Hajishirzi, Danqi Chen, and Luca Soldaini. 2025. Organize the Web: Constructing Domains Enhances Pre-Training Data Curation. In *Forty-second International Conference on Machine Learning*. https://openreview.net/forum?id=boSqwdvJVC

[75] Tingkai Yan, Haodong Wen, Binghui Li, Kairong Luo, Wenguang Chen, and Kaifeng Lyu. 2025. Larger Datasets Can Be Repeated More: A Theoretical Analysis of Multi-Epoch Scaling in Linear Regression. arXiv:2511.13421

[76] 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, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang,

Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 Technical Report. *CoRR* abs/2505.09388 (2025). arXiv:2505.09388 doi:10.48550/ARXIV.2505.09388

[77] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 Technical Report. *CoRR* abs/2407.10671 (2024). arXiv:2407.10671 doi:10.48550/ARXIV.2407.10671

[78] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 Technical Report. *CoRR* abs/2412.15115 (2024). arXiv:2412.15115 doi:10.48550/ARXIV.2412.15115

[79] Hu Yiwen, Huatong Song, Jie Chen, Jia Deng, Jiapeng Wang, Kun Zhou, Yutao Zhu, Jinhao Jiang, Zican Dong, Yang Lu, Xu Miao, Xin Zhao, and Ji-Rong Wen. 2025. YuLan-Mini: Pushing the Limits of Open Data-efficient Language Model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 5374–5400. doi:10.18653/v1/2025.acl-long.268

[80] Bowen Yu, Guanyu Feng, Huanqi Cao, Xiaohan Li, Zhenbo Sun, Haojie Wang, Xiaowei Zhu, Weimin Zheng, and Wenguang Chen. 2021. Chukonu: A Fully-Featured Big Data Processing System by Efficiently Integrating a Native Compute Engine into Spark. *Proc. VLDB Endow.* 15, 4 (2021), 872–885. doi:10.14778/3503585.3503596

[81] 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. *CoRR* abs/2501.08197 (2025). arXiv:2501.08197 doi:10.48550/ARXIV.2501.08197

[82] 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:2501.08197 [cs.CL] https://arxiv.org/abs/2501.08197

[83] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025. Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model?. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=4OsgYD7em5

[84] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauly, Michael J. Franklin, Scott Shenker, and Ion Stoica. 2012. Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. In *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2012, San Jose, CA, USA, April 25-27, 2012*, Steven D. Gribble and Dina Katabi (Eds.). USENIX Association, 15–28. https://www.usenix.org/conference/nsdi12/technical-sessions/presentation/zaharia

[85] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a Machine Really Finish Your Sentence?. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Anna Korhonen, David Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, Florence, Italy, 4791–4800. doi:10.18653/v1/P19-1472

[86] Biao Zhang and Rico Sennrich. 2019. Root Mean Square Layer Normalization. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 12360–12371. https://proceedings.neurips.cc/paper/2019/hash/1e8a19426224ca89e83cef47f1e7f53b-Abstract.html

[87] Yifan Zhang, Yifan Luo, Yang Yuan, and Andrew C Yao. 2025. Autonomous Data Selection with Zero-shot Generative Classifiers for Mathematical Texts. In *Findings of the Association for Computational Linguistics: ACL 2025*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 4168–4189. doi:10.18653/v1/2025.findings-acl.216

[88] Zhengyan Zhang, Yuxian Gu, Xu Han, Shengqi Chen, Chaojun Xiao, Zhenbo Sun, Yuan Yao, Fanchao Qi, Jian Guan, Pei Ke, Yanzheng Cai, Guoyang Zeng, Zhixing Tan, Zhiyuan Liu, Minlie Huang, Wentao Han, Yang Liu, Xiaoyan Zhu, and Maosong Sun. 2021. CPM-2: Large-scale cost-effective pre-trained language models. *AI Open* 2 (2021), 216–224. doi:10.1016/j.aiopen.2021.12.003

[89] Zhengyan Zhang, Xu Han, Hao Zhou, Pei Ke, Yuxian Gu, Deming Ye, Yujia Qin, Yusheng Su, Haozhe Ji, Jian Guan, Fanchao Qi, Xiaozhi Wang, Yanan Zheng, Guoyang Zeng, Huanqi Cao, Shengqi Chen, Daixuan Li, Zhenbo Sun, Zhiyuan Liu, Minlie Huang, Wentao Han, Jie Tang, Juanzi Li, Xiaoyan Zhu, and Maosong Sun. 2021. CPM: A large-scale generative Chinese Pre-trained language model. *AI Open* 2 (2021), 93–99. doi:10.1016/j.aiopen.2021.07.001

[90] Fan Zhou, Zengzhi Wang, Nikhil Ranjan, Zhoujun Cheng, Liping Tang, Guowei He, Zhengzhong Liu, and Eric P. Xing. 2025. MegaMath: Pushing the Limits of Open Math Corpora. *CoRR* abs/2504.02807 (2025). arXiv:2504.02807 doi:10.48550/ARXIV.2504.02807

[91] Kun Zhou, Beichen Zhang, Jiapeng Wang, Zhipeng Chen, Wayne Xin Zhao, Jing Sha, Zhichao Sheng, Shijin Wang, and Ji-Rong Wen. 2024. JiuZhang3.0: Efficiently Improving Mathematical Reasoning by Training Small Data Synthesis Models. (2024).

[92] Yutao Zhu, Kun Zhou, Kelong Mao, Wentong Chen, Yiding Sun, Zhipeng Chen, Qian Cao, Yihan Wu, Yushuo Chen, Feng Wang, Lei Zhang, Junyi Li, Xiaolei Wang, Lei Wang, Beichen Zhang, Zican Dong, Xiaoxue Cheng, Yuhan Chen, Xinyu Tang, Yupeng Hou, Qiangqiang Ren, Xincheng Pang, Shufang Xie, Wayne Xin Zhao, Zhicheng Dou, Jiaxin Mao, Yankai Lin, Ruihua Song, Jun Xu, Xu Chen, Rui Yan, Zhewei Wei, Di Hu, Wenbing Huang, Ze-Feng Gao, Yueguo Chen, Weizheng Lu, and Ji-Rong Wen. 2024. YuLan: An Open-source Large Language Model. *CoRR* abs/2406.19853 (2024). arXiv:2406.19853 doi:10.48550/ARXIV.2406.19853

Appendices

A Non-Embedding Based Comparison

In practice, the vocabulary sizes are different across different models, and embedding layers commonly account for relatively lower compute per parameter. We also note that the naming of different models has no consensus on using total parameters or non-embedding parameters in the model name. Hence, to conduct a more complete comparison, we also take statistics on both total parameters and non-embedding parameters of different open-weight and fully open-source models, and report the results in Table 4. In addition, taking non-embedding parameter as the X-axis, we report an additional comparison in Figure 8. We find that our model still excels the frontier of fully open-source models, and approaches close to leading open-weight models, like the Qwen series of a similar scale. Our advantage over other fully open-source models looks more prominent when taking account the non-embedding parameters.

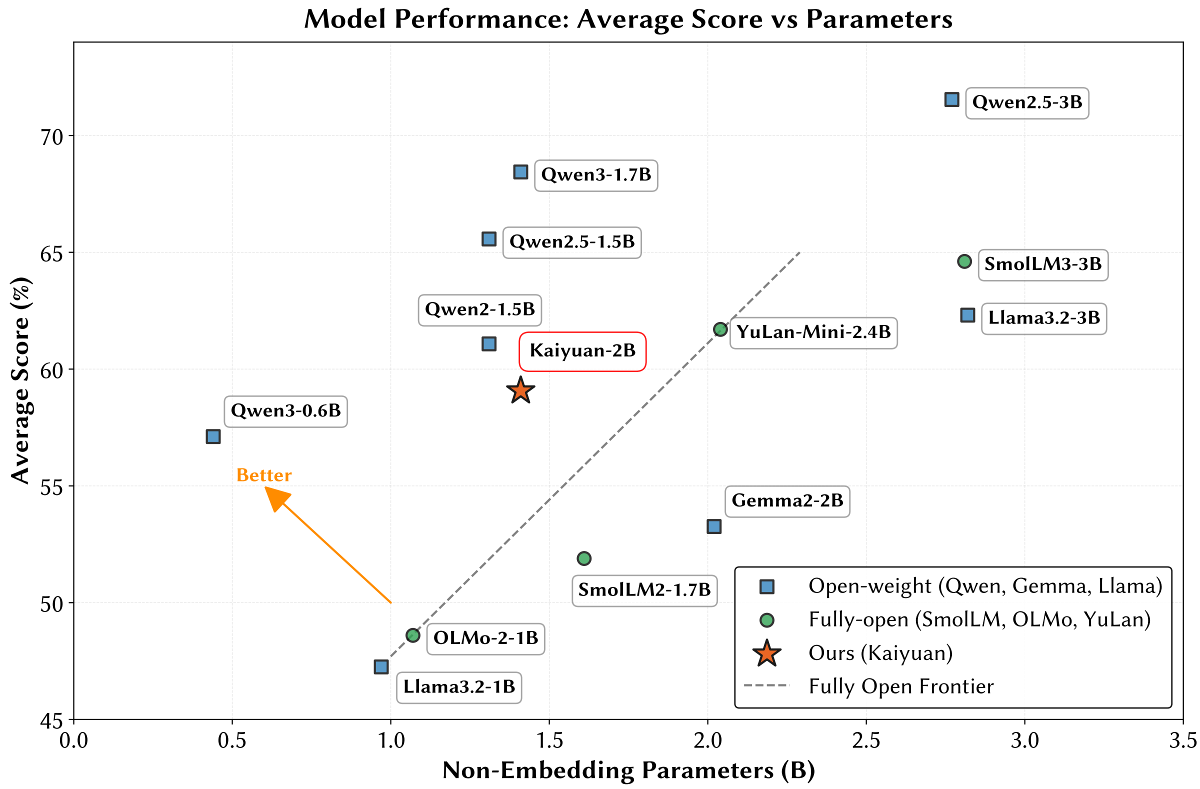


Figure 8: Model performance comparison over non-embedding parameters.

B Quality Score Quantile Benchmarking

We show full quantile benchmarking results in Figures 9 and 10. The overall observations are discussed in Section 3 in detail. The DCLM Baseline leading experiments are shown in Figure 9 and Fineweb-Edu leading experiments are shown in Figure 10.

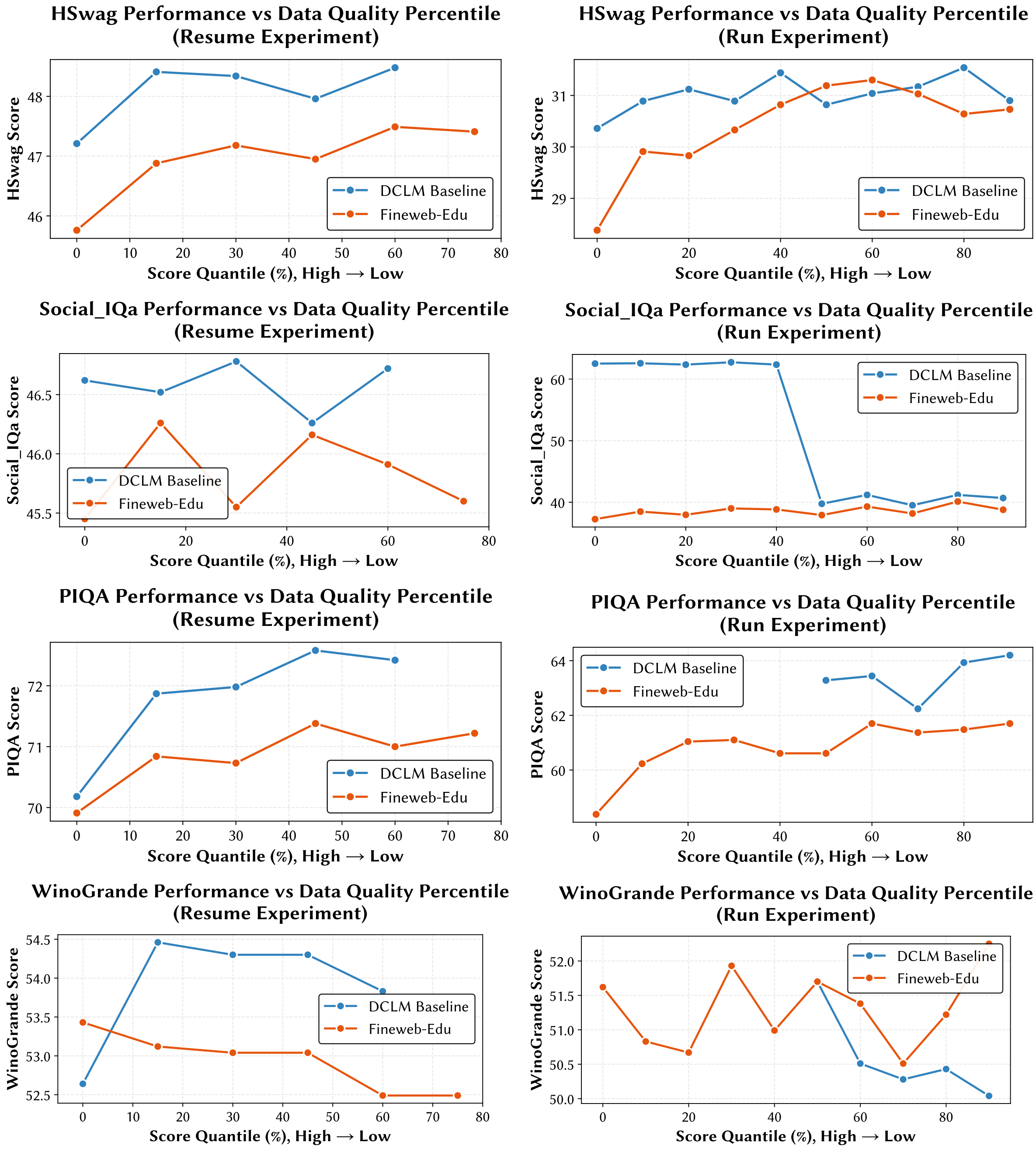


Figure 9: Quantile Benchmarks: DCLM Baseline is better on understanding-oriented benchmarks.

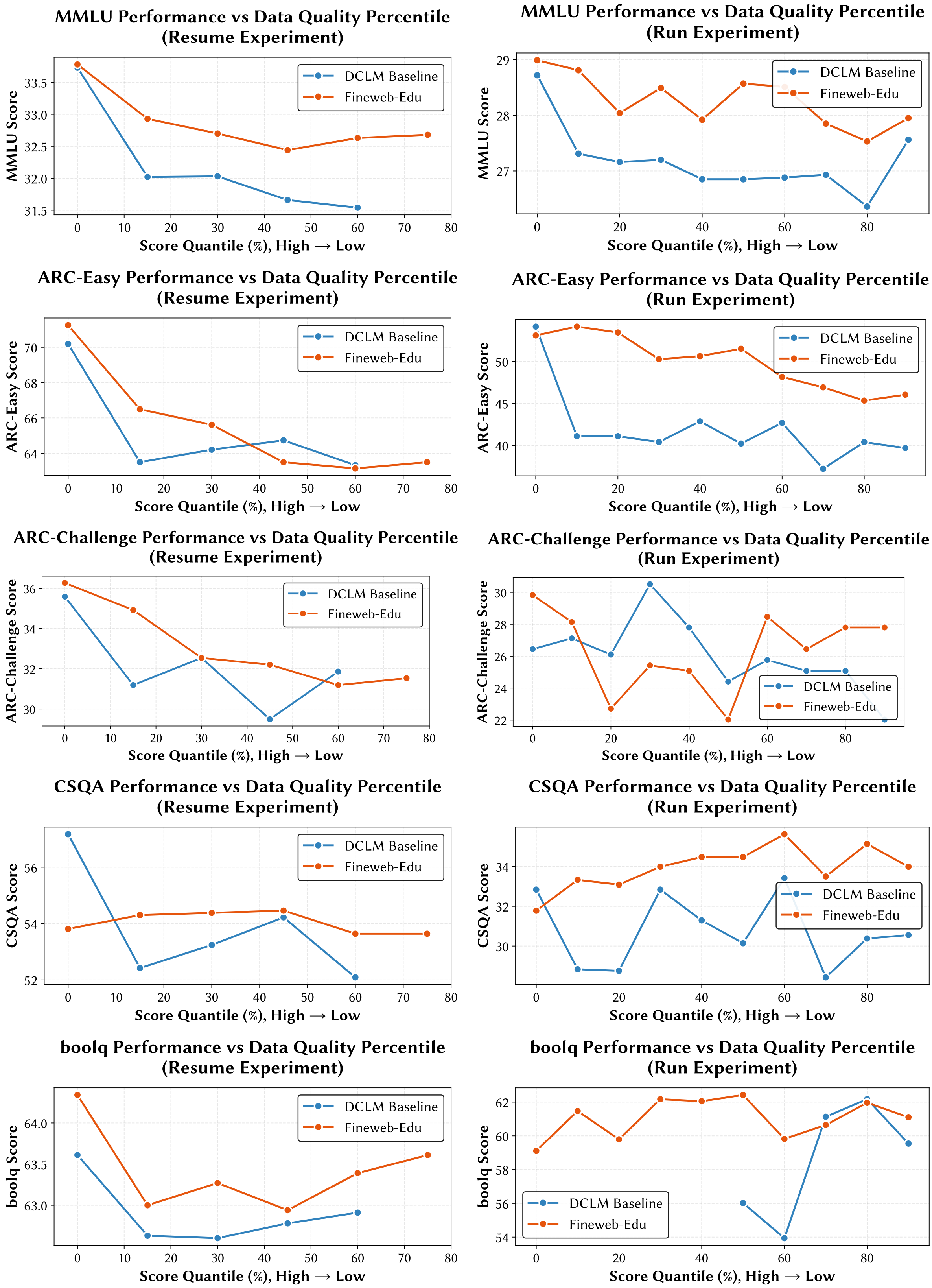


Figure 10: Quantile Benchmarks: FineWeb-Edu is better on knowledge-oriented benchmarks.

Table 4: Model Parameter Statistics Comparison

Model Name	Total	Embedding	Non-Embedding	Tied Embedding
SOTA Models				
Qwen2-1.5B	1.54B	0.23B	1.31B	TRUE
Qwen2.5-1.5B	1.54B	0.23B	1.31B	TRUE
Qwen2.5-3B	3.09B	0.31B	2.77B	TRUE
Qwen3-0.6B-Base	0.60B	0.16B	0.44B	TRUE
Qwen3-1.7B-Base	1.72B	0.31B	1.41B	TRUE
Qwen3-4B-Base	4.02B	0.39B	3.63B	TRUE
gemma-2-2B	2.61B	0.59B	2.02B	TRUE
Llama-3.2-1B	1.24B	0.26B	0.97B	TRUE
Llama-3.2-3B	3.21B	0.39B	2.82B	TRUE
Fully-Open SOTA Models				
SmolLM2-1.7B	1.71B	0.10B	1.61B	TRUE
OLMo-2-0425-1B	1.48B	0.41B	1.07B	FALSE
YuLan-Mini	2.42B	0.38B	2.04B	FALSE
SmolLM3-3B	3.08B	0.26B	2.81B	TRUE
Ours				
PCMind-2.1-Kaiyuan-2B	2.03B	0.62B	1.41B	FALSE

C Datasets Used in Training

Table 5 is a comprehensive list of all datasets used in the training process of PCMind-2.1-Kaiyuan-2B. All datasets are publicly available to acquire, and most of them are hosted on Hugging Face unless otherwise noted.

Table 5: All Datasets Used in the Training of PCMind-2.1-Kaiyuan-2B

Name	Type	Hugging Face ID	#Tokens[0]	License(s)
DCLM-Baseline	English	mlfoundations/dclm-baseline-1.0 [42]	4T	CC BY 4.0[1]
FineWiki-EN	English	HuggingFaceFW/finewiki [60]	8.7B	CC BY-SA 4.0[6]
FinePDFs	English	HuggingFaceFW/finepdfs [39]	3T	ODC-By 1.0[1]
Flan	English	allenai/dolmino-mix-1124	17B	ODC-By 1.0
Pes2O	English	allenai/dolmino-mix-1124	58.6B	ODC-By 1.0
FineWeb-Edu-EN	English	HuggingFaceTB/smollm-corpus [9]	220B	ODC-By 1.0[1]
ArXiv	English	togethercomputer/RedPajama-Data-1T [17]	28B	Metadata: CC0 1.0 [4] Content: various [3]
Cosmopedia-v2	English	HuggingFaceTB/smollm-corpus [9]	27B	ODC-By 1.0
FineWiki-CN	Chinese	HuggingFaceFW/finewiki [60]	1.1B	CC BY-SA 4.0[6]
Fineweb-Edu-CN	Chinese	opencsg/Fineweb-Edu-Chinese-V2.1 [82]	1.5T	OpenCSG Community License [16], Apache 2.0
Baidu-Baike	Chinese	mohamedah/baidu_baike	1.2B	MIT
UNDL ZH-EN Aligned	Chinese	bot-yaya/undl_zh2en_aligned	1.8B	MIT

Continued on next page

Table 5: All Datasets Used in the Training of PCMind-2.1-Kaiyuan-2B (Continued)

Name	Type	Hugging Face ID	#Tokens[0]	License(s)
Dedup-Merged-PAC-CN[4]	Chinese	BAAI/CCI-Data BAAI/CCI2-Data BAAI/CCI3-Data [72] Skywork/SkyPile-150B [73] OpenDataLab/WanJuan1.0 [27, 28][5] BAAI/IndustryCorpus BAAI/IndustryCorpus2 [65] WuDaoCorpus2.0 [88, 89][5]	178B	CCI{,2,3}-Data: CCI Usage Agreement [56] SkyPile-150B: Skywork Community License [66], Apache 2.0 WanJuan1.0: CC BY-4.0 IndustryCorpus{,2}: Apache 2.0 WuDaoCorpus2.0: Apache 2.0
OpenWebMath	Math	open-web-math/open-web-math [59]	14.7B	ODC-By 1.0[1]
FineMath	Math	HuggingFaceTB/finemath [2]	10B	ODC-By 1.0
MegaMath-Web-Pro	Math	LLM360/MegaMath [90]	300B	ODC-By 1.0
AutoMathText	Math	math-ai/AutoMathText [87]	8.7B	CC BY-SA 4.0
SwallowMath-v2	Math	tokyotech-llm/swallow-math-v2 [23]	32B	Apache 2.0
StarCoder	Code	bigcode/starcoderdata [38]	250B	Original Licenses[2]
Stack V2 Smol	Code	bigcode/the-stack-v2 [47]	900B	Original Licenses[2]
StackExchange	Code	togethercomputer/RedPajama-Data-1T [17]	20B	CC BY-SA 2.5/3.0/4.0[3]
Python-Edu	Code	HuggingFaceTB/smollm-corpus [47, 9]	3.4B	ODC-By 1.0, Original Licenses[2]
Algebraic-Stack	Code	typeof/algebraic-stack [6, 59]	11B	ODC-By 1.0[1]
Swallow-Code-v2	Code	tokyotech-llm/swallow-code-v2 [23]	49.8B	Apache 2.0
SlimOrca	SFT	Open-Orca/SlimOrca [54, 45]	190M	MIT
JiuZhang3.0-Corpus-CoT	SFT	ToheartZhang/JiuZhang3.0-Corpus-CoT [91]	358B	*Not Specified*
Tulu-3-Sft-0225	SFT	allenai/tulu-3-sft-mixture [40]	640M	ODC-By 1.0 (mixed)
downstream[4]	SFT	cais/mmlu [31, 30] openai/gsm8k [14] allenai/ai2_arc [13] allenai/openbookqa [51] Rowan/hellaswag [85] allenai/winogrande [63]	12.6M	MMLU, GSM8K: MIT ai2_arc: CC BY-SA 4.0 OpenBookQA: *Not Specified* hellaswag: MIT winogrande: *Not Specified*

[0] Token counts are pre-deduplication rough numbers. They may differ from the well-known ones due to partial inclusion of mixed datasets, the use of different revisions/splits/tokenizers, or some other pre-processing.
[1] This dataset originates from Common Crawl and thereby abides by its terms of use [15].
[2] This dataset contains source code with various licenses.
[3] The license has changed over time, according to `https://stackoverflow.com/help/licensing`.
[4] This dataset is created by mixing and de-duplicating all source datasets.
[5] This dataset is acquired from OpenDataLab (`https://opendatalab.com`).
[6] Some old content of Wikipedia is dual-licensed under CC BY 4.0 and GFDL.

To enhance the reproducibility of our results and accessibility, we have conducted careful screening and selection of datasets at the best of our ability. We would like to ensure that our model (Kaiyuan-2B) and training datasets are compliant with all licenses and agreements presented in table 5, thus can be released under a permissive license for the community to use (still on an "as-is" and "use-at-your-own-risk" basis). Everyone could use these same datasets to reproduce our results, and further adapt and/or publish both the modified datasets and models at will, free of any legal risk.

For example, although the Nemotron series datasets from NVIDIA are also available on Hugging Face upon request, the *NVIDIA Data Agreement for Model Training* [55] applied to them disallows redistribution, and even public display of the dataset. Therefore, they are fully excluded from our training data.

D Phase-wise Data Mixture

In this section, we first visualize the dataset counts within each domain throughout multi-phase training. The transitions of the English, Chinese, Math, Code, and SFT datasets are shown in Figure 11, Figure 12, Figure 14, Figure 13, and Figure 15, respectively. Moreover, we list the detailed dataset composition for each phase in Tables 6 to 10, from Phase 1 to Phase 5. In these tables, there are four primary cases:

1. The entire dataset is used in this phase. The score column is denoted as *(fully used)*, and the actual ratio is 100.0%, such as DCLM-Baseline in Phase 1 (Table 6) and Fineweb-Edu-EN in Phase 2 (Table 7).
2. The dataset is filtered according to its specific score column (*Score Col* in the tables), retaining only top-scoring samples with an *Actual Ratio*. For example, Fineweb-Edu-CN in Phase 1 keeps the top 20.8% of *score* (Table 6), and StarCoder in Phase 2 keeps the top 10.4% of *max_stars_count*.
3. The dataset has no quality metrics, and we randomly select samples accounting for the *Actual Ratio*. For example, we randomly select 10.0% of samples from StarCoder and 30.0% from LLM360-Math in Phase 1 (Table 6).
4. The dataset is repeated within the phase. The score column is denoted as *duplicate*, and the actual ratio exceeds 100%. The repetition count is determined by rounding the actual ratio according to its decimal part. For example, FineWiki-CN is repeated twice in Phase 3 (Table 8), and for Baidu-Baike in Phase 5, we round 1.5 to either 1 or 2 with equal probability, then repeat the samples that many times (Table 10).

In addition, LLM360-Math is a deduplicated subset of the MegaMath dataset [90], and we select only the top 5% of rows from the English partition of the FinePDFs dataset [39], according to Fineweb-Edu classifier scores [61].

Table 6: Phase 1 Dataset Statistics

Dataset	Score Col	Token Before (B)	Token After (B)	Actual Ratio
DCLM-Baseline	(fully used)	608.54	608.54	100.0%
FineWeb-Edu-CN	score	441.66	91.78	20.8%
StarCoder	random	190.60	19.08	10.0%
LLM360-Math	random	31.12	9.34	30.0%

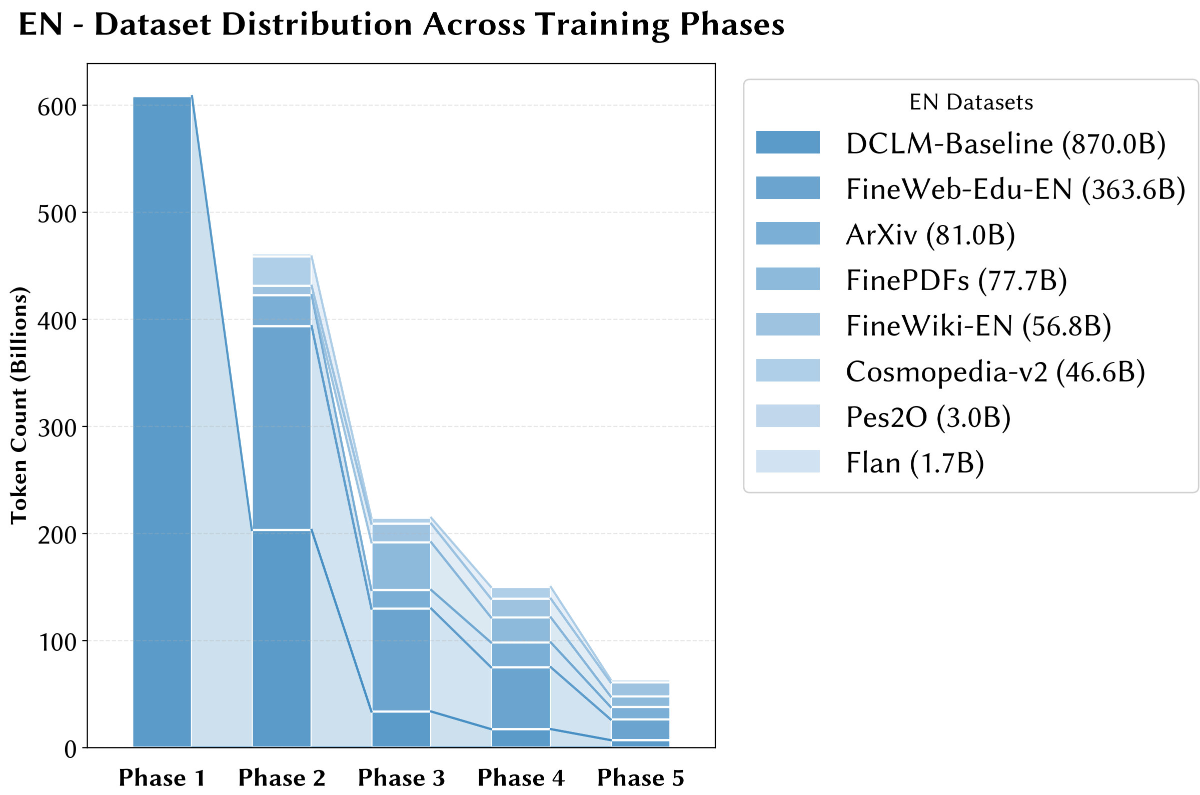


Figure 11: Phase-wise dataset mixture for English.

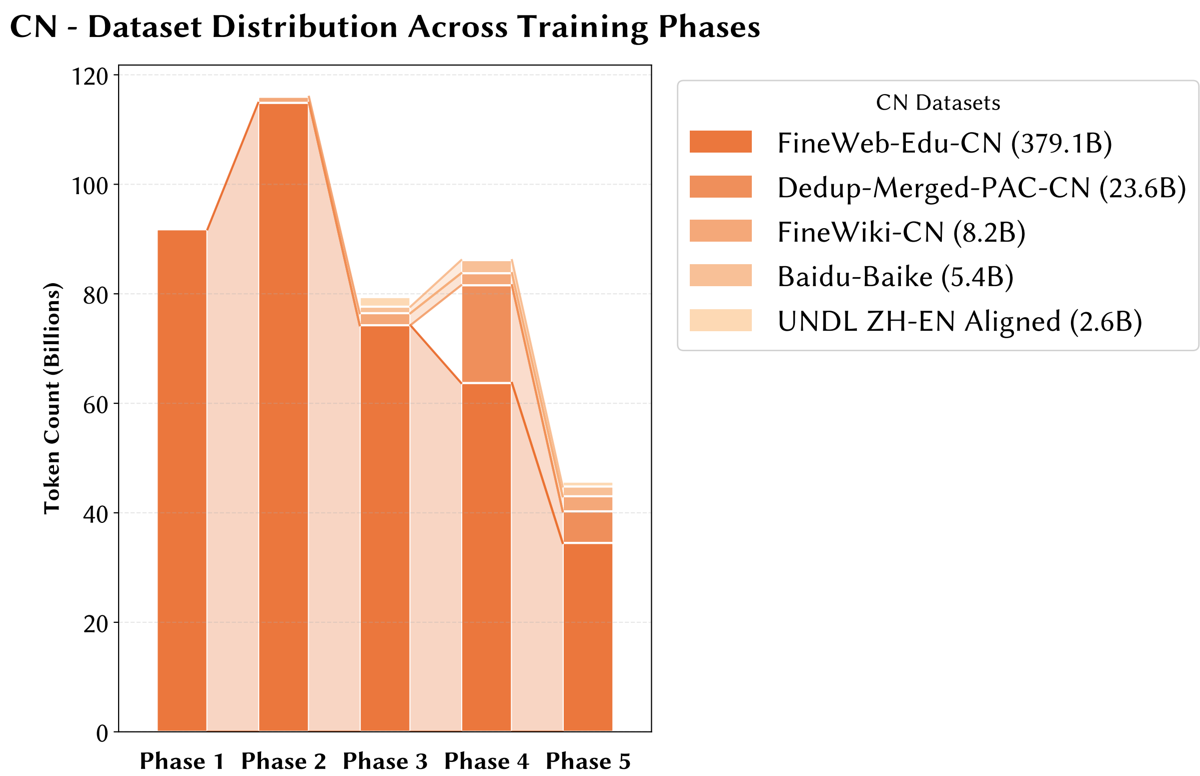


Figure 12: Phase-wise dataset mixture for Chinese.

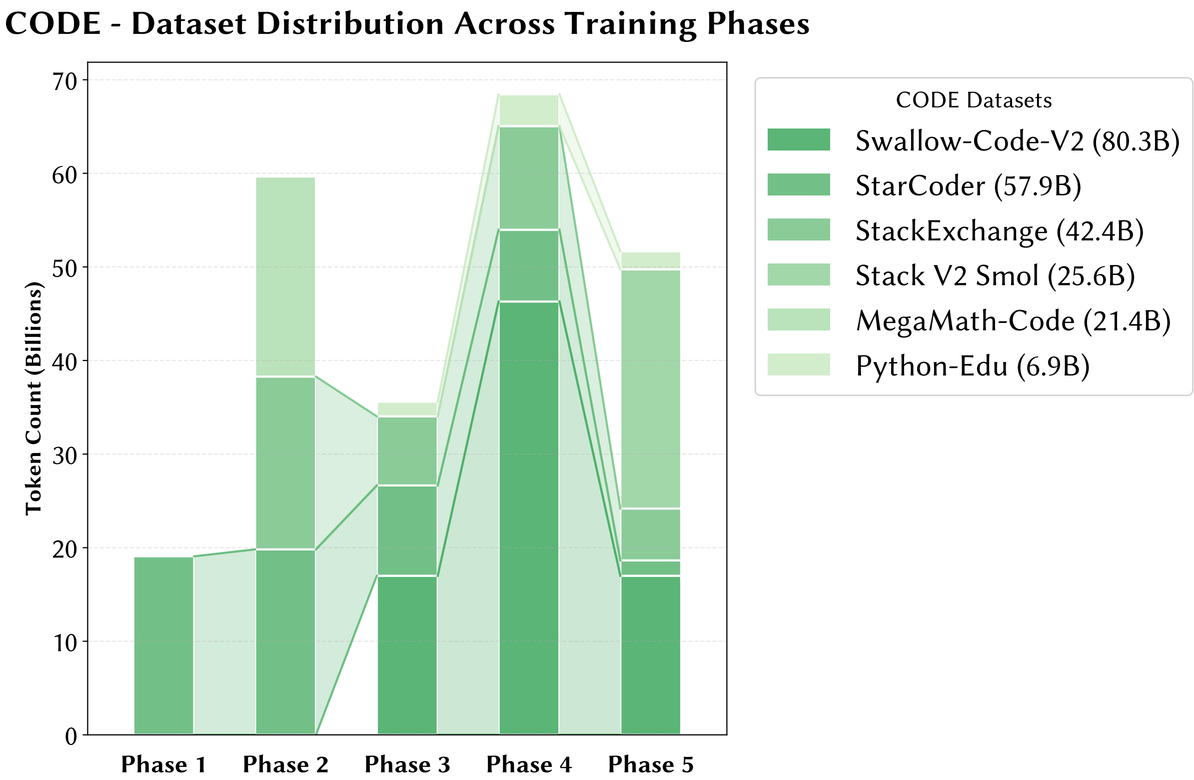


Figure 13: Phase-wise dataset mixture for Code.

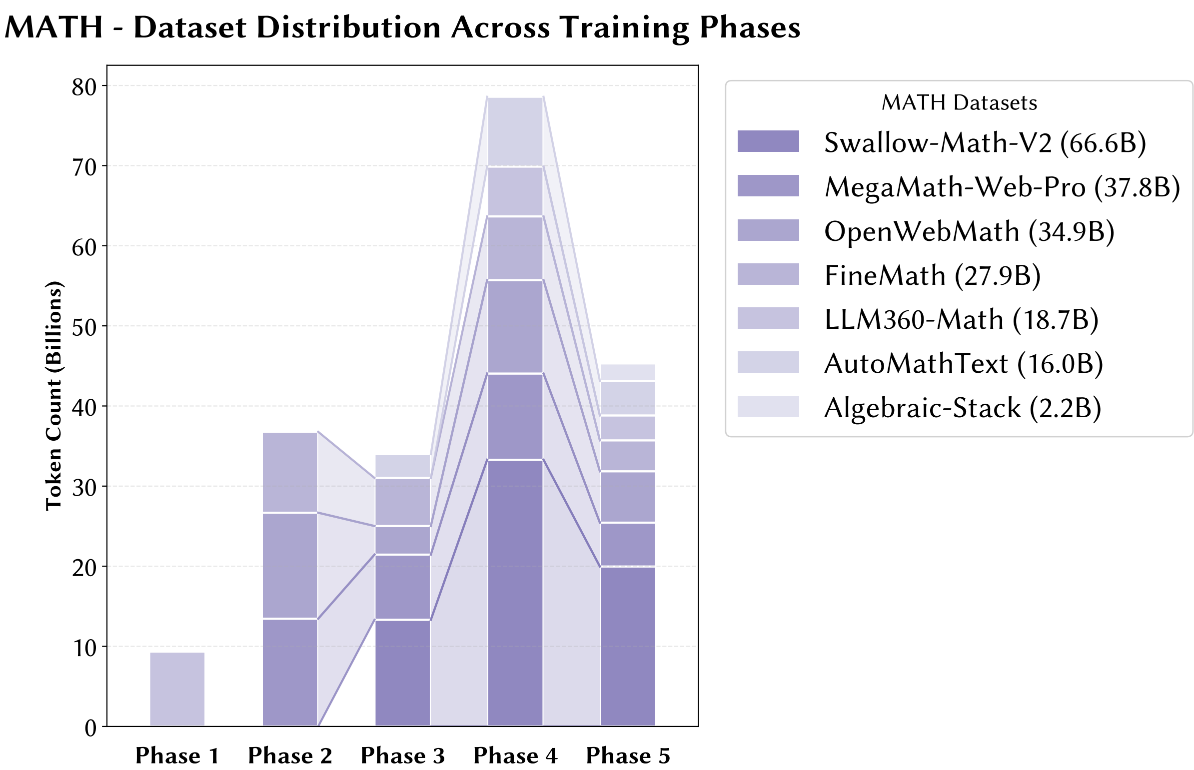


Figure 14: Phase-wise dataset mixture for Math.

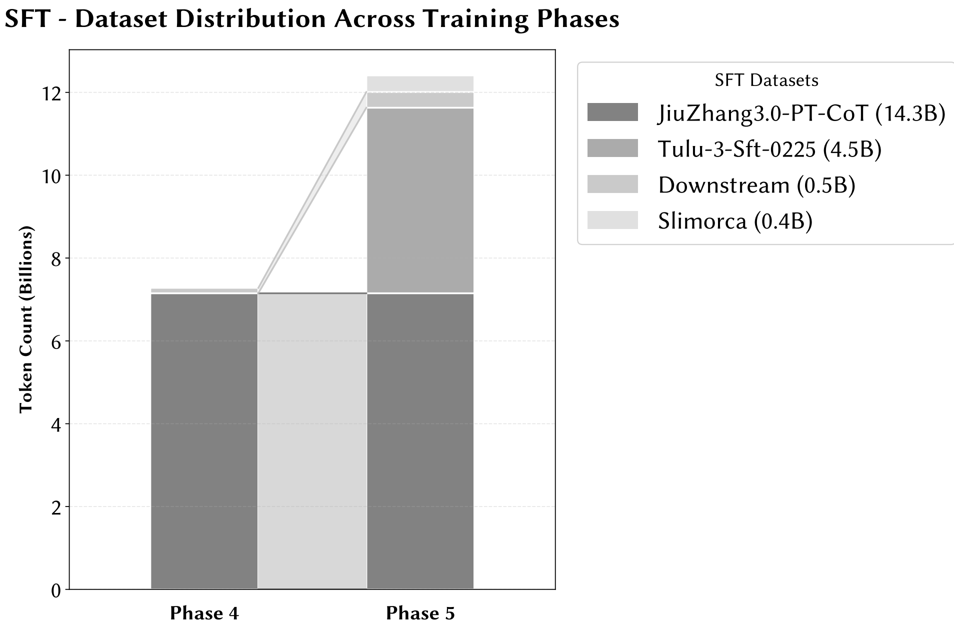


Figure 15: Phase-wise dataset mixture for SFT.

Table 7: Phase 2 Dataset Statistics

Dataset	Score Col	Token Before (B)	Token After (B)	Actual Ratio
FineWeb-Edu-CN	score	441.66	114.88	26.0%
FineWiki-CN	(fully used)	1.10	1.10	100.0%
FineWeb-Edu-EN	(fully used)	190.37	190.37	100.0%
DCLM-Baseline	fasttext score	608.54	203.32	33.4%
Flan	random	17.15	1.71	10.0%
Pes2O	random	60.11	3.00	5.0%
FineWiki-EN	(fully used)	8.74	8.74	100.0%
ArXiv	(fully used)	28.93	28.93	100.0%
Cosmopedia-v2	(fully used)	27.41	27.41	100.0%
FineMath	(fully used)	10.10	10.10	100.0%
OpenWebMath	(fully used)	13.23	13.23	100.0%
MegaMath-Web-Pro	(fully used)	13.45	13.45	100.0%
StackExchange	(fully used)	18.46	18.46	100.0%
MegaMath-Code	random	42.77	21.38	50.0%
StarCoder	max_stars_count	190.60	19.82	10.4%

Table 8: Phase 3 Dataset Statistics

Dataset	Score Col	Token Before (B)	Token After (B)	Actual Ratio
FineWeb-Edu-CN	score	441.66	74.26	16.8%
FineWiki-CN	duplicate	1.10	2.20	200.0%
UNDL ZH-EN Aligned	(fully used)	1.75	1.75	100.0%
Baidu-Baike	(fully used)	1.19	1.19	100.0%
FineWeb-Edu-EN	score	190.37	96.13	50.5%
DCLM-Baseline	fasttext score	608.54	33.77	5.5%
FineWiki-EN	duplicate	8.74	17.47	200.0%
ArXiv	random	28.93	17.35	60.0%
FineMath	score	10.10	6.00	59.4%
MegaMath-Web-Pro	math_score	13.45	8.13	60.4%
StackExchange	random	18.46	7.38	40.0%
StarCoder	max_stars_count	190.60	9.65	5.1%
Swallow-Code-V2	score	50.62	17.00	33.6%
Python-Edu	score	3.41	1.56	45.7%
Cosmopedia-v2	random	27.41	5.48	20.0%
AutoMathText	lm_q1q2_score	8.71	2.97	34.1%
OpenWebMath	math_score	13.23	3.57	27.0%
Swallow-Math-V2	random	33.29	13.32	40.0%
FinePDFs	(fully used)	44.50	44.50	100.0%

Table 9: Phase 4 Dataset Statistics

Dataset	Score Col	Token Before (B)	Token After (B)	Actual Ratio
FineWeb-Edu-CN	score	441.66	63.71	14.4%
FineWiki-CN	duplicate	1.10	2.20	200.0%
Baidu-Baike	duplicate	1.19	2.39	200.0%
FineWeb-Edu-EN	score	190.37	57.79	30.4%
DCLM-Baseline	fasttext score	608.54	17.32	2.8%
FineWiki-EN	duplicate	8.74	17.47	200.0%
ArXiv	random	28.93	23.15	80.0%
FineMath	score	10.10	7.95	78.7%
MegaMath-Web-Pro	math_score	13.45	10.76	80.0%
StackExchange	random	18.46	11.07	60.0%
StarCoder	max_stars_count	190.60	7.67	4.0%
Downstream	duplicate	0.01	0.13	1000.0%
Swallow-Code-V2	score	50.62	46.30	91.5%
Python-Edu	(fully used)	3.41	3.41	100.0%
Cosmopedia-v2	random	27.41	10.97	40.0%
AutoMathText	(fully used)	8.71	8.71	100.0%
LLM360-Math	random	31.12	6.22	20.0%
OpenWebMath	math_score	13.23	11.66	88.1%
Swallow-Math-V2	(fully used)	33.29	33.29	100.0%
JiuZhang3.0-PT-CoT	duplicate	3.58	7.15	200.0%
FinePDFs	fineweb-edu-classifier	44.50	23.38	52.5%
Dedup-Merged-PAC-CN	random	178.49	17.85	10.0%

Table 10: Phase 5 Dataset Statistics

Dataset	Score Col	Token Before (B)	Token After (B)	Actual Ratio
FineWeb-Edu-CN	score	441.66	34.50	7.8%
FineWiki-CN	duplicate	1.10	2.75	250.0%
UNDL ZH-EN Aligned	random	1.75	0.88	50.3%
Baidu-Baike	duplicate	1.19	1.79	150.0%
FineWeb-Edu-EN	score	190.37	19.35	10.2%
DCLM-Baseline	fasttext score	608.54	7.06	1.2%
FineWiki-EN	duplicate	8.74	13.10	150.0%
ArXiv	random	28.93	11.60	40.1%
FineMath	score	10.10	3.86	38.2%
MegaMath-Web-Pro	math_score	13.45	5.47	40.7%
StackExchange	random	18.46	5.54	30.0%
StarCoder	max_stars_count	190.60	1.64	0.9%
Downstream	duplicate	0.01	0.38	3000.0%
Swallow-Code-V2	score	50.62	17.00	33.6%
Python-Edu	score	3.41	1.92	56.3%
Cosmopedia-v2	random	27.41	2.74	10.0%
AutoMathText	lm_q1q2_score	8.71	4.32	49.6%
LLM360-Math	random	31.12	3.11	10.0%
OpenWebMath	math_score	13.23	6.41	48.5%
Swallow-Math-V2	random	33.29	19.97	60.0%
JiuZhang3.0-PT-CoT	duplicate	3.58	7.15	200.0%
FinePDFs	fineweb-edu-classifier	44.50	9.86	22.2%
Dedup-Merged-PAC-CN	pac_score	178.49	5.77	3.2%
Tulu-3-Sft-0225	duplicate	0.64	4.48	700.0%
Stack V2 Smol	random	127.98	25.56	20.0%
Slimorca	duplicate	0.20	0.40	200.0%
Algebraic-Stack	max_stars_count	8.51	2.17	25.5%

E Experimental Settings

E.1 Implementation of Stability Components

To maintain numerical values within the FP16 safety margin without sacrificing model performance, we implement Logits Soft-Capping and Sandwich Normalization. These mechanisms cap extreme values and normalize residual branches, respectively.

Logits Soft-Capping Standard linear layers in Large Language Models often produce logits that grow unbounded during training, causing the Softmax function to saturate and gradients to vanish or explode. Soft-capping addresses this by squashing the logits into a fixed range using the hyperbolic tangent (tanh) function before scaling them back. Formally, given the raw logits x and a capping threshold σ (e.g., 30.0 or 50.0), the capped logits x' are computed as:

$$x' = \sigma \cdot \tanh\left(\frac{x}{\sigma}\right) \tag{1}$$

In our implementation, we apply this transformation to the output logits of the language model head. This ensures that the input to the cross-entropy loss remains within the range $(-\sigma, \sigma)$, preventing logits from exceeding the FP16 maximum value while preserving the relative order of probabilities.

Sandwich Normalization In the standard Pre-Norm Transformer architecture, the input x is normalized before the sub-layer (Attention or Feed-Forward Network), and the output is added directly to the residual stream: $x_{l+1} = x_l + F(\text{Norm}(x_l))$. While effective, this allows the magnitude of the residual stream x to grow monotonically with depth, potentially destabilizing deep networks. Sandwich Normalization introduces an additional normalization layer explicitly on the output of the sub-layer branch before the residual addition. The modified update rule for a block containing a sub-layer F (e.g., Self-Attention or MLP) is defined as:

$$x_{l+1} = x_l + \text{Norm}_{\text{post}}\left(F(\text{Norm}_{\text{pre}}(x_l))\right) \tag{2}$$

In our implementation, we apply this strictly to the residual branches. This ensures that the contribution of each layer has unit variance, preventing the accumulation of extreme activation values as the network depth increases.

E.2 Training Configuration

In Table 11, we present the details of our training hyperparameter configuration in three parts:

- For the model architecture, we primarily follow Qwen3-1.7B [76] and adopt the vocabulary from the Qwen series [77, 76]. We use $\theta = 10000$ for RoPE [68] to support a context length of 4K. The Soft-Capping threshold is set to 30.0, as discussed in Section 2.
- We use AdamW as the optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.95$. We adopt a μP with base dimension of 896 and set the learning rate to 5×10^{-3} for Phase 1 and 3×10^{-3} thereafter before decay.
- To support FP16 training, we use dynamic loss scaling with a factor of 2 and a window of 20 to handle widely varying gradient scales.

This detailed configuration facilitates the reproduction of our training run.

E.3 Model Average

Following recent work [48], we average the near-end checkpoints to reduce variance and consolidate learned knowledge and capabilities. We first evaluate the last eight checkpoints on a subset of lightweight benchmarks, as shown in Table 12. Consecutive checkpoints are spaced 400 steps apart, corresponding to 3.36B tokens. These checkpoints fluctuate during training and do not exhibit a clear upward or downward trend. Therefore, we apply simple model averaging [44], directly averaging the last eight checkpoints to obtain the final model.

E.4 Reference Experiments for Quantile Benchmarking

We conduct quantile benchmarking experiments across two primary scenarios: training from scratch and continual training from checkpoints. For each experiment, given a target quantile p%, we select the data partition above the p% threshold, comprising approximately 10B tokens, which are then used for the respective training scenarios.

Table 11: Training Hyperparameter Configuration

Category	Parameter	Value
Model Architecture		
	Sequence Length	4096
	Hidden Size	2048
	FFN Dimension	6144
	Number of Layers	28
	Number of Attention Heads	16
	Number of KV Heads (GQA)	8
	Vocabulary Size	151936
	Rotary θ	10000.0
	Logit Soft-capping threshold	30.0
	Initialization Std	0.018
Optimizer Configuration		
	Optimizer Type	AdamW
	Learning Rate (Phase 1)	5×10^{-3}
	Learning Rate (Phase 2+)	3×10^{-3}
	Batch Size	2048
	β_1	0.9
	β_2	0.95
	ϵ	1e-8
	Weight Decay	0.1
	Warmup Steps	5000
	μP Width Base	896
Loss Scaling (Dynamic)		
	Scale Factor	2
	Scale Window	20
	Minimum Loss Scale	524288

Table 12: Model Performance Across Checkpoints

Checkpoint Step	ARC-Challenge	ARC-Easy	CSQA	PIQA	Average
260632	64.41	82.72	65.93	73.39	71.61
261032	65.42	82.19	65.36	73.72	71.67
261432	63.05	81.48	65.68	74.59	71.2
261832	65.42	81.83	64.78	73.56	71.40
262232	61.36	83.25	65.77	73.78	71.04
262632	65.76	82.01	66.34	73.5	71.90
263032	63.73	80.78	66.42	73.78	71.17
263132	62.71	80.6	66.09	73.88	70.82

Training from Scratch. In the training-from-scratch scenario, we train a model with the Qwen3-0.6B architecture [76]. Following the default configuration in Table 11, we conduct a small-scale experiment using the settings detailed in Table 13, training over approximately 8.4B tokens from the quantile data chunks. We employ a constant learning rate schedule with a sufficiently long warmup phase to ensure stable training dynamics.

Continual Training. In the continual training scenario, we resume from a checkpoint previously trained on approximately 367B tokens of the deduplicated DCLM Baseline dataset. The model adopts the Qwen2.5-0.5B architecture [78]. We then train over approximately 8.4B tokens from the quantile data chunks using the configuration specified in Table 14. For these experiments, we linearly decay the learning rate from a peak value of 1×10^{-3} to a final value of 1×10^{-5}.

Consistency Across Scenarios. As illustrated in Figures 9 and 10, the benchmarking results exhibit strong alignment between the training-from-scratch and continual training experiments. This consistency persists for evaluations on both the DCLM Baseline and Fineweb-Edu datasets, despite resuming from a checkpoint trained exclusively on the deduplicated DCLM Baseline dataset. This observation supports the robustness of our quantile-based data selection approach across different training paradigms.

Table 13: Training Hyperparameter Configuration for Quantile Benchmarking: Training from Scratch

Parameter	**Value**
Learning Rate	1×10^{-3}
Batch Size	512
Warmup Steps	400
Total Steps	4000

Table 14: Training Hyperparameter Configuration for Quantile Benchmarking: Continual Training

Parameter	**Value**
Peak Learning Rate	1×10^{-3}
Final Learning Rate	1×10^{-5}
Batch Size	2048
Total Steps	1000

E.5 Reference Experiments for Repetition and Curriculum Model Averaging

These experiments primarily follow the experimental framework established in CMA [48]. We use a model with the Qwen2.5-1.5B architecture without tied embeddings and train on a subset of the first shard of the DCLM-Baseline dataset.

Baseline Configuration. The baseline experiment adopts uniform data ordering and employs a Warmup-Stable-Decay (WSD) learning rate schedule with a 1-sqrt decay function [26, 71], decaying to a near-zero final learning rate. The detailed experimental configuration is provided in Table 15.

High-Quality Data Utilization Strategies. To investigate effective high-quality data utilization, we explore two complementary approaches:

(1) **Repetition Strategy:** We repeat high-quality data partitions for various top-k retention ratios, matching the computational FLOPs of the single-pass baseline experiment for fair performance comparison.

(2) **Curriculum with Model Averaging:** We adopt CMA/CDMA[3] [48], which integrates curriculum learning with either no or moderate LR decay, accompanied by model averaging over the final checkpoints.

[3] We do not distinguish these variants in our context, and refer to both as *CMA*. By definition, the CMA method in Table 1 corresponds to the CDMA variant, which retains LR decay.

Table 15: Training Hyperparameter Configuration for Baseline and Repetition

Parameter	**Value**
Peak Learning Rate	3×10^{-3}
Final Learning Rate	1×10^{-5}
Batch Size	512
Total Steps	15,375
Decay Steps	2,875
Warmup Steps	768

Table 16: Training Hyperparameter Configuration for Curriculum Model Average

Parameter	**Value**
Peak Learning Rate	3×10^{-3}
Final Learning Rate	1×10^{-3}
Batch Size	512
Total Steps	15,375
Decay Steps	2,875
Warmup Steps	768
Checkpoint Number	6
Decay Factor of EMA (α)	0.2
Checkpoint Interval	0.21B

Experimental Variants. The repetition experiments follow identical settings to the baseline, differing only in dataset construction. For the curriculum experiments, we use a higher final learning rate of 1×10^{-3} and perform an exponential moving average (EMA) over the final six checkpoints (the last-step checkpoint is weighted by $(1 - \alpha)$ relative to the current-step checkpoint, where α is the decay factor), replicating the methodology from CMA [48].

Evaluation Settings. In Table 1, we evaluate performance on a high–signal-to–noise-ratio benchmark subset (*Core* in Table 1) comprising MMLU [31], ARC [13], and CSQA [69], following established practices in prior work [29, 48]. These benchmarks provide strong discriminative power for identifying performance differences between training approaches.

F Model Performance across Benchmarks (Full Table)

Table 17 merges the results from both Tables 2 and 3, providing a complete evaluation of the models across all target capability dimensions. Because the models differ in total parameters and non-embedding parameters, we present performance–parameter visualizations in Figures 1 and 8. These plots show that Kaiyuan-2B lies on the frontier of fully open-source models.

Table 17: Comparison of Model Performance across Various Benchmarks

Model Name	Params	Math		Code		Chinese		Reasoning & Knowledge									Avg.
		GSM8K	MATH	sanitized_MBPP	HumanEval	C-Eval	CMMLU	MMLU	ARC-C	ARC-E	BoolQ	CSQA	HSwag	PIQA	SocIQ	Wino	
Open-Weight SOTA Models																	
Qwen2-1.5B	1.5B	58.50	21.70	50.58	31.10	71.29	70.62	56.36	70.17	83.60	71.90	70.52	60.77	75.73	63.46	59.83	61.08
Qwen2.5-1.5B	1.5B	68.50	35.00	58.37	37.20	68.63	68.01	61.56	79.32	90.48	76.39	75.10	64.18	76.17	64.94	59.67	65.57
Qwen2.5-3B	3B	79.10	42.60	66.54	42.10	74.65	73.92	66.86	86.44	92.59	83.88	76.09	73.85	81.45	69.40	63.69	71.54
Qwen3-0.6B	0.6B	59.59	32.44	51.75	29.88	57.03	52.36	55.09	68.14	84.48	69.05	61.18	48.51	69.97	61.51	55.64	57.11
Qwen3-1.7B	1.7B	75.44	43.50	64.20	52.44	66.70	66.55	65.35	80.34	91.89	79.82	74.61	60.76	77.20	68.58	59.27	68.44
Qwen3-4B	4B	87.79	54.1	74.32	62.2	78.5	77.01	75.78	89.83	97.53	86.09	81.9	79.46	84.98	75.59	65.43	78.03
gemma2-2B	2B	23.90	15.00	38.91	17.70	41.35	39.63	55.20	66.44	82.54	72.42	69.45	66.20	78.89	65.92	65.35	53.26
llama-3.2-1B	1B	44.40	30.60	34.63	18.90	29.82	31.03	37.74	36.95	70.55	67.43	62.82	60.20	74.92	50.61	58.17	47.25
llama-3.2-3B	3B	77.70	48.00	49.42	29.88	45.67	44.33	57.87	72.20	83.95	76.73	70.35	71.06	79.05	64.33	64.09	62.31
Fully-Open SOTA Models																	
SmolLM2-1.7B	1.7B	31.10	11.60	49.42	22.60	35.06	34.03	51.99	59.66	82.72	69.85	67.16	65.30	78.51	60.18	59.12	51.89
OLMo-2-0425-1B	1B	68.30	20.70	15.56	6.71	30.53	28.62	44.25	47.46	76.72	70.55	65.60	61.61	76.44	55.53	60.38	48.60
YuLan-Mini-2.4B	2.4B	66.65	27.12	62.26	61.60	52.32	48.14	51.76	64.75	82.54	78.59	66.18	61.20	77.31	63.25	61.88	61.70
SmolLM3-3B	3B	67.63	46.10	62.26	39.63	50.84	49.35	63.04	77.29	88.54	76.12	70.52	69.20	79.05	65.25	64.40	64.61
Ours																	
PCMind-2.1-Kaiyuan-2B	2B	51.33	30.34	56.42	42.68	46.30	49.25	53.90	66.10	82.89	78.53	67.40	58.13	74.37	62.59	65.75	59.07