KAIYUAN-2B: Pushing the Limits of Fully-Open Language Model through Data Benchmarking and Curriculum

Anonymous ACL submission

Abstract

The rapid advancement of Large Language Models (LLMs) has resulted in a significant 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 **KAIYUAN-2B**, a fully open-source 2-billion-parameter model focused on improving training efficiency and effectiveness under resource constraints. Our methodology includes three key innovations in a data-centric way: a *Quantile Data Benchmarking* method for systematically comparing heterogeneous open-source datasets and providing insights on data mixing strategies; a *Bi-Level Curriculum Training* policy that progressively introduces domain-specialized and refined samples at both phase and instance levels; and a *Strategic Selective Repetition* scheme within the multi-phase paradigm to effectively leverage sparse, high-quality data. KAIYUAN-2B achieves performance competitive with state-of-the-art fully open-source models, demonstrating practical and scalable solutions for resource-limited pretraining. We release all assets (including model weights, data, and code) under the Apache 2.0 license.[1]

1 Introduction

Scaling the volume and quality of pretraining data, as well as model scale, has driven the rapid advancement of Large Language Models (LLMs). However, the core engineering "recipes" for state-of-the-art models remain largely opaque. The community currently faces two major transparency challenges: (1) the pretraining of proprietary models is closed-source (OpenAI, 2023; Gemini Team, 2025); and (2) some models are released with open weights but with closed-source training recipes (DeepSeek-AI et al., 2024; Yang et al., 2025).

Academic progress relies on fully open-source initiatives that release weights, data, and detailed methodologies. While pioneers like OLMo (Groeneveld et al., 2024) and SmolLM (Allal et al., 2025a) have made significant strides, matching the performance of industry-leading open-weight models remains difficult under academic resource constraints. In this report, we introduce **KAIYUAN-2B**, a fully open-source model designed to narrow this performance gap through data-centric innovations. We specifically address two critical bottlenecks: **(1) Heterogeneous Open-Source Data:** Existing datasets (Li et al., 2024c; Penedo et al., 2024) use varied preprocessing and scoring pipelines, making it difficult to systematically compare and mix them effectively. **(2) Limited Compute Resources:** Academic groups cannot typically match the tens of trillions of tokens used for training by leading industry labs, underscoring the need for greater training efficiency and more strategic data utilization.

To tackle these bottlenecks, we propose practical solutions centered on data utilization: benchmarking heterogeneous open-source datasets, phase-wise and instance-wise curriculum learning, and strategic selective repetition of high-quality data. KAIYUAN-2B advances the Pareto frontier of the performance-parameter trade-off among fully open-source models (Figure 1), with 1.4B non-embedding parameters trained over 2.2T tokens.

Our primary contributions are:

Quantile Data Benchmarking. Due to the absence of a unified metric to compare quality scores across different open-source datasets, we introduce *quantile benchmarking* to provide a unified evaluation of heterogeneous datasets systematically. To achieve this, we first fix several quality-score quan-

[1]The model weight and dataset are released in Huggingface but we do not include links here for the double-blind review policy. The data preprocessing framework is in https://anonymous.4open.science/r/Kaiyuan-Spark-4E83/.

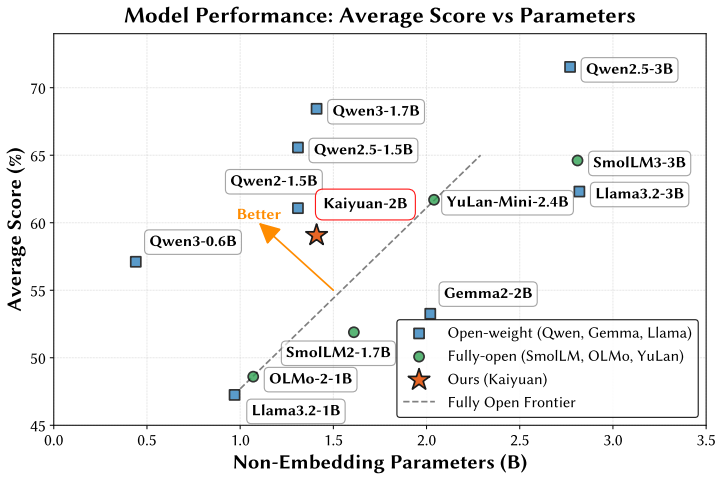


Figure 1: **Performance over non-embedding parameters.** KAIYUAN-2B surpasses the frontier of fully open-source models at a similar scale and narrows the gap to leading open-weight models like Qwen2-1.5B. See Table 2 for full results. Parameter counts for various models are provided in Table 9.

tiles, then quantify how data at different score levels contributes to specific model capabilities by training small reference models using data around these quantiles. This method deepens the understanding of heterogeneous dataset properties, especially for leading open-source datasets, and enables data selection and mixing guided by quantitative results (§ 3).

Bi-Level Curriculum Training. We adopt curriculum training at two levels. First, we conduct a 5-phase training process that progressively increases domain data (Chinese, code, math) ratios phase by phase. Second, we incorporate a quality-based instance-level curriculum in the final three stages, where data are presented in ascending order of quality. To ensure effective learning of the most informative samples, we adopt a moderate learning rate (LR) decay and apply model averaging, based on Curriculum Model Average (CMA) (Luo et al., 2025). The use of this larger learning rate than usual enables the model to learn high-quality data sufficiently (§ 4).

Strategic Selective Repetition. Recognizing that high-quality data is scarce, we employ *selective repetition* on high-quality data classified by our quantile data benchmarking. Unlike typical pretraining, which visits each data sample only once, our setup allows high-quality samples to be revisited across phases—up to a 5-phase limit—to maximize the utility of limited high-quality tokens without overfitting (§ 5).

Parameter Efficiency Frontiers. KAIYUAN-2B attains great parameter efficiency, pushing the limit of fully-open models. KAIYUAN-2B establishes a new performance-parameter Pareto optimal for fully open-source models around the 2B scale (Figure 1).[2] It substantially outperforms the open-weight Gemma2-2B in math and coding tasks (Table 10) and matches its general abilities with fewer parameters. It also matches the average performance of the larger Yulan-2.4B in reasoning and knowledge tasks (Table 11). Moreover, KAIYUAN-2B is trained on 2.2T tokens, focusing more on data quality than merely quantity, and exhibits great data and compute efficiency, thereby facilitating the open-source community. Comparisons based on total parameters are in Figure 14, and parameter details are in Table 9. (§ 7)

2 Related Works

Recent advancements in fully open-source pretraining are exemplified by the OLMo series (Walsh et al., 2025; Muennighoff et al., 2025), SmolLM series (Allal et al., 2025a; Bakouch et al., 2025), Nemotron series (Basant et al., 2025; NVIDIA et al., 2025), and YuLan series (Hu et al., 2024; Zhu et al., 2024). These initiatives enhance transparency by disclosing weights, pre-training corpora, training recipes, and data processing frameworks (Soldaini et al., 2024; Kuchaiev et al., 2019).

Proprietary data mixtures and undisclosed training recipes remain the primary causes of the performance gap between open-source models and industry leaders. While open-weight models like the Qwen series (Yang et al., 2024a,b, 2025) release detailed architectural designs, the open-source community still struggles with the efficiency of data strategies. Current fully open-source models typically follow two paradigms. First, models such

[2] Since vocabulary sizes vary and embedding layers require less compute per parameter, we use non-embedding parameters for the X-axis.

as OLMo2-1B and SmolLM2-1.7B adopt an "over-trained" regime, utilizing 4T and 11T tokens respectively. This extreme token-to-parameter ratio imposes prohibitive computational and data costs, hindering iterative research and replication. Second, data-efficient alternatives like YuLan-Mini (Yiwen et al., 2025) utilize fewer tokens (e.g., 1.1T) but require complex human intervention, involving over 20 training stages and manually adjusted sampling ratios.

Both paradigms suffer from a lack of principled methodology, as the rationale behind their data recipes remains largely heuristic or manually handcrafted. This limitation is exacerbated by the recent surge of diverse, high-volume open-source datasets (Yu et al., 2025a; Su et al., 2025; Li et al., 2024c), because there are few effective and inexpensive testbeds to evaluate these datasets and draw heuristics from. We argue that bridging the performance gap requires a deeper understanding of data quality and its integration with training strategies.

Our work addresses these hurdles through two primary contributions: (1) a **quantile benchmarking** approach to evaluate datasets across multiple dimensions rigorously, and (2) a **data-aware training paradigm** that utilizes selective repetition and interleaved multi-dataset curricula to optimize efficiency. Our final model, featuring 1.4B non-embedding parameters trained on 2.2T tokens, establishes a replicable benchmark for academic pre-training research.

3 Data Quantile Benchmarking

Constructing high-quality pretraining corpora from heterogeneous open-source data presents two fundamental challenges: (1) the absence of a unified metric to compare quality across disparate sources (e.g., DCLM Baseline (Li et al., 2024c) vs. FineWeb-Edu (Penedo et al., 2024)), and (2) the prohibitive cost of exhaustive ablation to determine optimal mixing ratios. To address these, we propose **Quantile Data Benchmarking**, a compute-efficient framework to evaluate dataset characteristics across quality distributions empirically.

One approach to evaluate the quality score metric is by performing top-k filtering based on this metric and then evaluating the models trained on the filtered data (Li et al., 2024c; SHUM et al., 2025; Mizrahi et al., 2025). However, this approach only reflects average quality above a threshold and fails to capture the entire data quality distribution. Quantile Benchmarking instead stratifies a dataset into quality-score quantiles and probes them independently. This method not only benchmarks the target dataset, but it also simultaneously reflects the intrinsic inclinations of its corresponding score metrics.

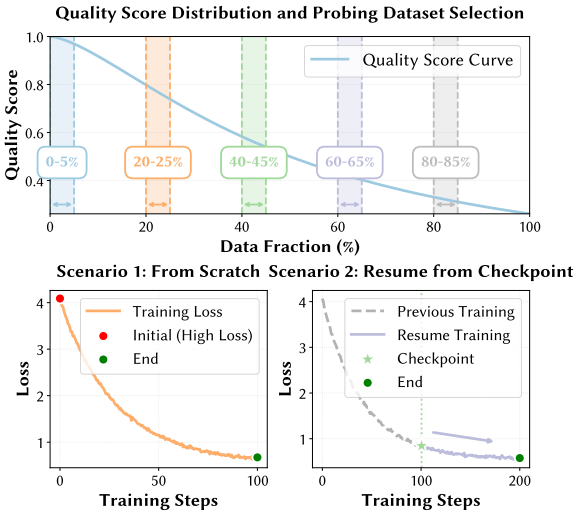


Figure 2: Quantile Benchmarking Framework. We extract fixed-size probing chunks (e.g., 5% of total tokens) from specific quality quantiles (e.g., top 0%, 20%, ...). Small-scale reference models (0.5B) are then trained to evaluate each chunk's utility for specific downstream capabilities, covering both training-from-scratch and resume-from-checkpoint scenarios.

Methodology. Our quantile benchmarking framework is illustrated in Figure 2. Given a dataset with associated quality scores, we first identify a target quantile set Q (e.g., $\{0, 0.1, \ldots, 0.8\}$). For each $q \in Q$, we select a probing subset $\mathcal{D}_q$ of fixed-size tokens (e.g., 5B) starting from that percentile. We then train a reference model (0.5B parameters) under two regimes: (1) **From-scratch**, to measure raw data utility, and (2) **Continual training**, to assess the data as a refinement signal. Finally, we evaluate the resulting checkpoints across a suite of benchmarks.

This approach is highly efficient. First of all, the quantile benchmarking is primarily conducted on dominant datasets, like DCLM Baseline, FineWeb-Edu-Chinese-v2.1, etc., since they account for the majority of pretraining data but are not extensively understood. For the DCLM-Baseline (609B tokens after deduplication), benchmarking five quantiles requires only 42B tokens, utilizing 8.4B tokens per quantile. Using a 0.6B reference model, the total computational cost is less than 0.6% of the total KAIYUAN-2B pretraining budget. This cost

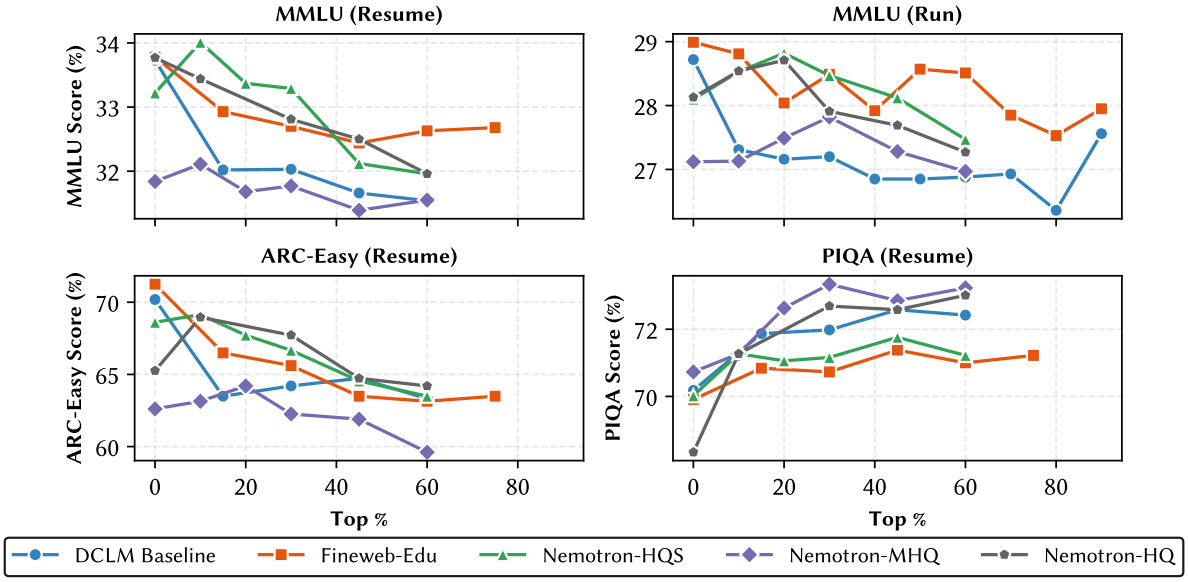


Figure 3: Representative results from quantile benchmarking. (1) Task-dependent dataset characteristics: FineWeb-Edu and Nemotron-HQS excel on knowledge-intensive tasks (MMLU, ARC-Easy), whereas DCLM-Baseline and Nemotron-MHQ perform better on commonsense reasoning (PIQA). (2) Internal heterogeneity: Performance on ARC-Easy can differ by more than 8% between the best and worst quantile partitions within FineWeb-Edu. (3) Biased refinement: Prevailing score metrics align with knowledge-oriented tasks (MMLU, ARC-Easy) but can be less informative or even show reverse correlation in other dimensions (PIQA). (4) Regime consistency: Both from-scratch and resume-from-checkpoint setups show similar inclinations. For instance, in MMLU, DCLM Baseline and Nemotron-MHQ underperform overall in both settings, though relative rankings may not be identical.

can be amortized among future pretraining while providing more granular insights than traditional scaling law ablations.

Empirical Findings. Benchmarking dominant English datasets—DCLM Baseline (Li et al., 2024c), FineWeb-Edu (Penedo et al., 2024), and Nemotron-CC-v2 (Su et al., 2025; Basant et al., 2025)—yields key insights into leading open-source corpora and their quality metrics[3] (Figure 3). We perform global deduplication on DCLM Baseline and FineWeb-Edu, retaining approximately one-fifth of the original volume[4]. For Nemotron-CC-v2, duplication ratios are significantly lower; for example, the high-quality-synthetic partition does not exceed 5%. Consequently, we directly benchmark their partitions, including high-quality (Nemotron-HQ), high-quality-synthetic (Nemotron-HQS), and medium-high-quality (Nemotron-MHQ). Downstream benchmarks evaluate general knowledge (e.g., MMLU (Hendrycks et al., 2021b), ARC (Clark et al., 2018)) and commonsense understanding and reasoning (e.g., PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019)). Representative results are shown in Figure 3, with full results in Figures 7 and 8.

Task-Dependent Superiority: Datasets exhibit domain-specific strengths that reflect the intrinsic inclinations of their quality metrics. As shown in Figure 3, FineWeb-Edu and Nemotron-HQ excel in knowledge-intensive tasks, while DCLM-Baseline shows a consistent advantage in situated commonsense reasoning. This dependency likely stems from metric design: DCLM-Baseline is refined toward datasets like ELI-5, which emphasizes accessible explanations, whereas FineWeb-Edu is refined based on educational value (Li et al., 2024c; Penedo et al., 2024). These findings explicitly reveal the intrinsic characteristics of the datasets and are partially testified by evaluation results reported in prior work (Wettig et al., 2025). Moreover, they offer clear guidance for future experimental design: *for knowledge-intensive tasks (e.g., MMLU), models should be trained on FineWeb-Edu or Nemotron-CC-v2, whereas for understanding-oriented tasks (e.g., HellaSwag), training on the DCLM Baseline is more appropriate.*

Internal Heterogeneity: Substantial performance variance exists within individual datasets. In Figure 3, for the DCLM dataset, MMLU perfor-

[3]DCLM Baseline corresponds to the DCLM fasttext score; FineWeb-Edu corresponds to the FineWeb-Edu classifier. For Nemotron-CC-v2, which does not disclose quality metrics, we annotate samples using the PreSelect score (SHUM et al., 2025) via a lightweight fasttext model.

[4]For FineWeb-Edu, we use the FineWeb-Edu-Dedup partition from the SmolLM corpus (Ben Allal et al., 2024)

mance drops by 2% between the top 0% and 60% quantiles, and ARC-Easy performance drops by 8% in resume experiments and even up to 15% in the from-scratch setup (Figure 8). Similar patterns are also observed for other datasets. Such heterogeneity necessitates quality-aware selection even within supposedly "clean" sources.

Biased Refinement and Non-Monotonicity: Most data quality metrics correlate well with knowledge-oriented capabilities. For example, MMLU and ARC-Easy scores show a consistent trend across quantiles (Figures 3 and 8). However, higher quality scores do not always yield better performance. Paradoxically, across all evaluated metrics and datasets, higher scores lead to worse performance on understanding- and reasoning-oriented tasks, like PIQA (Figures 3 and 7). This non-monotonicity highlights the risk of over-filtering and the task-specific nature of data "quality," suggesting a need for more systematic, unbiased quality measurements.

Regime Consistency: The relative ranking of data partitions remains stable whether training from scratch or continuing from a checkpoint. This consistency holds across different datasets and quality metrics (Figures 7 and 8). The resume checkpoint was sufficiently pretrained on the DCLM Baseline dataset (details in Section E.4). This observation suggests that future efforts can focus on resume experiments, as they provide reliable downstream accuracy trends at a lower cost.

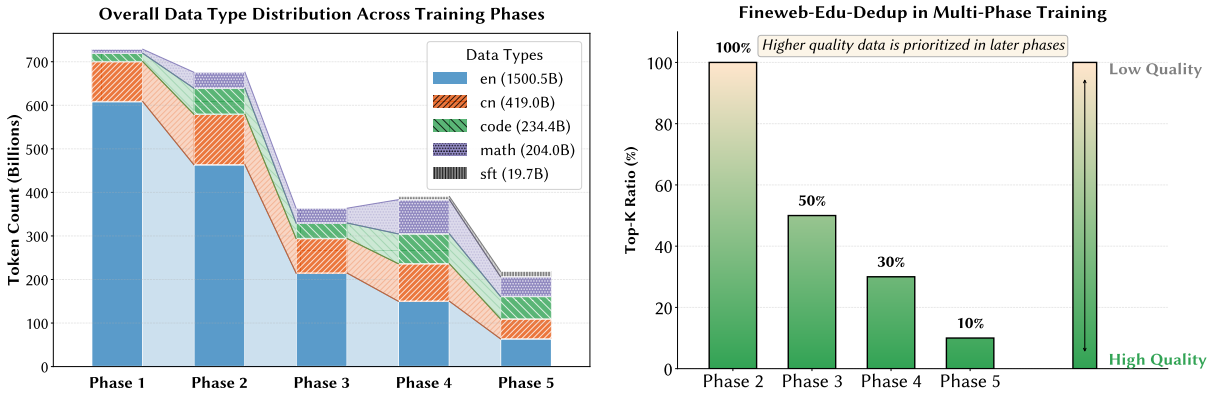
Implications for Selection. These findings directly inform our training strategy (Sections 4 and 5). First, we employ benchmark-guided ratios to balance different datasets and enhance model capabilities. Since quality metrics correlate reliably with knowledge-intensive tasks like MMLU, we use the downstream performance of reference models as a unified metric to align heterogeneous datasets. For example, our results (Figure 3) show that only the top 30% of the DCLM Baseline matches the utility of FineWeb-Edu and Nemotron high-quality partitions in the MMLU dimension; performance greatly drops below this quantile. We therefore use these performance anchors to determine selection thresholds. As a result, shown in Table 14, the top 30% of DCLM Baseline and full FineWeb-Edu datasets are mixed in the second phase. Second, we implement a **quality-based curriculum**, scheduling the highest-performing quantiles for the final training stages to maximize convergence on high-utility tokens. Finally, despite their high utility, we exclude Nemotron partitions from our final recipe to ensure strict compliance with licensing constraints (Section C).

4 Multi-Phase and Instance-Level Curriculum

We implement a multi-phase pretraining strategy to address the data quality heterogeneity identified in Section 3. While high-quality samples enhance model capabilities more efficiently than average data, they represent only a small fraction of the available tokens. To maximize data efficiency, we adopt curriculum training at both the phase and instance levels. Specifically, we progressively increase the **proportions** of specialized domain data (Chinese, Mathematics, and Code) in successive phases. Simultaneously, we implement an instance-level curriculum that prioritizes refined samples by increasing their relative ratios across phases and introduces an instance-level interleaved curriculum within the last three phases, respectively. This dual-level scheduling ensures that the model's final convergence is focused on the most informative and domain-specific data, leading to superior performance-parameter trade-offs. The data preprocessing framework to support our data-centric methods is introduced in Section E.6.

Phase-Level Curriculum. We partition the pretraining process into five distinct phases (Figure 4). This progression involves a gradual shift in the domain mixture: early phases focus on general-purpose English corpora, while later phases introduce specialized domains (Chinese, Code, and Math) and supervised finetuning (SFT) samples. To maintain stability during these transitions, we keep the English part at least 30% of the mixture in each phase, ensuring English remains the dominant language across phases. Details are in Section D.

Instance-Level Curriculum. Implementing a curriculum across heterogeneous datasets is challenging because source corpora often utilize incompatible quality metrics. We address this with a three-step interleaving algorithm (Algorithm 1). First, we perform **Within-Dataset Ranking** by sorting samples in each dataset D_i by local quality scores. Second, we apply **Rank Rescaling** to align dataset scales, computing a rescaled global rank $R(x) = r_i(x) \times (N_{\text{total}}/|D_i|)$, where $r_i(x)$ is the local rank and N_{total} is the combined sample



(a) Phase-wise data mixture transitions

(b) Phase-wise top-k ratio for FineWeb-Edu

Figure 4: The five training phases of KAIYUAN-2B. Later phases prioritize increasingly refined data samples.

count. Finally, we execute **Global Interleaving** by merging all datasets and sorting the union by $R(x)$. This ensures a stable global mixture ratio throughout the phase while allowing the average batch quality to increase monotonically.

To leverage this curriculum, we incorporate **Curriculum Model Averaging (CMA)** (Luo et al., 2025). This approach alleviates the conflict between curriculum data and the low learning rates (LR) typical of the late part of training. By using a moderate LR decay alongside model averaging, we maximize the utility of high-quality data while reducing optimization noise. In our setup, the LR is decayed to 6×10^{-4} (20% of the peak LR at 3×10^{-3}) in the final phase, where we average the last eight checkpoints to stabilize weights on the high-quality tail of the distribution.

Ablation results in Table 1 demonstrate that CMA improves average scores (50.99 vs. 50.56) compared to uniform sampling. Performance gains are particularly notable in the "Core" set[5], which consists of high-signal benchmarks used to distinguish model capabilities (Heineman et al., 2025). The experiment details are in Section E.5.

5 Multi-Phase Progressive Data Repetition

We further improve the efficacy of high-quality data through **Quality-Based Selective Repetition**. While repetition is discussed in prior work like D4 (Tirumala et al., 2023), it has not been extensively scaled for general pretraining. Our strategy utilizes a quality-based exposure mechanism. We revisit high-utility partitions across phases, aligned with the multi-phase setup, with the highest-quality

[5]including MMLU (Hendrycks et al., 2021b), ARC-Easy and ARC-Challenge (Clark et al., 2018), CSQA (Talmor et al., 2019)

Algorithm 1 Multi-Dataset Curriculum Construction

Require: Datasets $D_1, D_2, \ldots, D_k$ with dataset-specific quality metrics
Ensure: Multi-dataset curriculum dataset
$N_{\text{total}} \leftarrow \sum_{i=1}^{k} |D_i|$ ▷ Total sample count
for each dataset D_i **do**
  (Optional) Assign random scores for datasets lacking quality labels
  Sort D_i by quality metric in ascending order ▷ Within-dataset ranking
  Assign ordinal ranks $r_i(x) \in [1, |D_i|]$ to each sample $x \in D_i$
  Compute rescaled ranks: $R(x) \leftarrow r_i(x) \times \frac{N_{\text{total}}}{|D_i|}$
end for
$U \leftarrow \bigcup_{i=1}^{k} D_i$ ▷ Merge all datasets
Sort U by $R(x)$ in ascending order ▷ Global interleaving
return sorted U

partitions receiving the most exposure to amplify their impact on final capabilities. Unlike standard multi-epoch pretraining (Muennighoff et al., 2023; Yan et al., 2025)—repeat the dataset as a whole—we progressively decrease the retention ratio $k\%$ of datasets across phases, in a more fine-grained way. For example, with FineWeb-Edu (Figure 4), we utilize the full dataset in Phase 2, then retain only the top 50%, 30%, and 10% in subsequent phases. Consequently, the highest-quality 10% of samples are seen four times, while the lowest-quality samples appear only once.

Ablation experiments with a 1.5B-parameter model on a 30B token budget validate the efficiency of this approach. As shown in Table 1, the `Filter&Repeat` strategy (retaining the top 33.4% for three epochs) outperforms uniform one-pass training, confirming that repeating refined subsets improves parameter efficiency in data-constrained scenarios (Muennighoff et al., 2023). However, overly aggressive repetition (e.g., repeating the top 13.8% seven times) can cause overfitting, improving MMLU while degrading average performance.

Table 1: Selective Repetition and Curriculum Learning Strategies Ablation, Compared to Uniform Baseline.

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

In our setup, the repetitions are mostly capped to the number of training phases to prevent such degradation.

6 Training Configuration and Design

We train our 2B-parameter model using an architecture based on Qwen3-1.7B (1.4B non-embedding, 0.3B embedding parameters), omitting tied embeddings to reduce the communication overhead of shared parameters. The model is trained with a context length of 4,096 and a batch size of 2,048. We employ a Warmup-Stable-Decay schedule with a peak LR of 5×10^{-3}, reduced to 3×10^{-3} after Phase 1 to mitigate instability from inter-phase distribution shifts. Training is conducted on Ascend 910A clusters. Since these clusters lack BF16 support, we utilize FP16 along with sandwich normalization and soft clipping to ensure numerical stability (Section A). Detailed settings for the primary training and ablation experiments are provided in Section E.

7 Evaluation

We conduct extensive evaluations of KAIYUAN-2B against both open-weight and fully open-source models. Our results demonstrate that KAIYUAN-2B advances the parameter-efficiency frontier for fully open models, significantly narrowing the gap with leading proprietary-recipe counterparts and exhibiting great compute-efficiency.

7.1 Evaluation Setup

7.1.1 Baseline Models

We compare KAIYUAN-2B against state-of-the-art baselines with comparable parameter counts, categorized into two groups: **open-weight models** (public weights with proprietary data/recipes) and **fully open models** (public weights, architecture, code, and datasets). All evaluations use base checkpoints without finetuning.[6]

[6]For consistency, we standardize naming by omitting suffixes; e.g., "Qwen3" denotes the base model, regardless of its HuggingFace designation.

Open-weight models. This group includes **Qwen2-1.5B** (Yang et al., 2024a) (7T tokens), optimized for multilingual and coding tasks; the **Qwen2.5 series** (Yang et al., 2024b) (1.5B and 3B, 18T tokens), featuring refined architectures for mathematical reasoning; and the **Qwen3 series** (Yang et al., 2025) (0.6B to 4B, 36T tokens), supporting extended contexts. We also include **Gemma2-2B** (Rivière et al., 2024), distilled using 2T tokens, and the **Llama3.2 series** (Meta AI, 2024) (1B and 3B, 9T tokens), designed for on-device inference.

Fully open models. We compare against **SmolLM2-1.7B** (Allal et al., 2025a) (11T tokens), which utilizes the Llama 2 architecture; **SmolLM3-3B** (Bakouch et al., 2025), a data-centric model trained on 11T tokens; **OLMo2-1B** (Walsh et al., 2025) (4T tokens); and **YuLan-Mini** (Yiwen et al., 2025), a 2.4B-parameter model optimized for data efficiency using 1.1T tokens, with handcrafted fine-grained 28-phase training.

7.1.2 Benchmarks

Evaluation spans four primary domains. **Mathematics** is assessed via GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021c). **Coding** proficiency is measured using the MBPP (Austin et al., 2021) sanitized subset and HumanEval (Chen et al., 2021). **Chinese language processing** utilizes CMMLU (Li et al., 2024a) and C-Eval (Huang et al., 2023). Finally, **General Reasoning & Knowledge** is assessed via a suite of eight benchmarks: MMLU (Hendrycks et al., 2021b), HellaSwag (Zellers et al., 2019), CSQA (Talmor et al., 2019), BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), SocialIQA (Sap et al., 2019), WinoGrande (Sakaguchi et al., 2020), and ARC (Clark et al., 2018).

7.1.3 Implementation Details

Evaluations are conducted using the OpenCompass framework (Contributors, 2023). Mathematics and

Table 2: Comprehensive comparison across all benchmarks. KAIYUAN-2B shows great parameter-efficiency among fully open-source models.

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																	
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

coding tasks use *generation mode*[7], while other benchmarks employ *perplexity-based (PPL) evaluation*. Following the OLMES protocol (Gu et al., 2025), PPL tasks are assessed under both multiple-choice (MCF) and completion formulations (CF), with the higher score reported.

7.2 Evaluation Results

Performance summaries are provided in Tables 10 and 11, with full results in Table 2.

Core Capabilities: Math, Code, and Chinese (Table 10). KAIYUAN-2B achieves an average score of `46.05` across specialized benchmarks, outperforming similarly scaled fully open models like SmolLM2-1.7B. On Chinese tasks, KAIYUAN-2B (C-Eval: `46.30`; CMMLU: `49.25`) markedly exceeds OLMo2-1B and rivals the larger SmolLM3-3B. In mathematics, KAIYUAN-2B achieves `30.34` on MATH, surpassing YuLan-Mini-2.4B (`27.12`). In coding, KAIYUAN-2B reaches `42.68` on HumanEval, outperforming both SmolLM3-3B (`39.63`) and Qwen2.5-3B (`42.10`), demonstrating superior parameter efficiency.

Reasoning and Knowledge (Table 11). KAIYUAN-2B attains an average score of `67.74` across nine reasoning benchmarks, matching YuLan-Mini-2.4B (`67.50`) despite using fewer parameters. It surpasses SmolLM2-1.7B by 1.69 points within the fully open category. Moreover, KAIYUAN-2B performs competitively with Gemma2-2B (`69.16`) using a comparable token count, while larger open-weight models like Qwen3-4B maintain a lead due to significantly larger training budgets.

[7]For generation tasks, we report official results for baseline models when available, as exact reproduction can be challenging.

Parameter-Efficient and Compute-Economic. KAIYUAN-2B consistently advances the performance frontier for fully open-source models, outperforming similarly sized models such as Gemma2-2B and SmolLM2-1.7B while rivaling larger baselines like YuLan-Mini-2.4B and Llama3.2-3B (Figure 1, with comprehensive evaluation results are provided in Table 2). This positioning shows extraordinary parameter efficiency.

Beyond parameter count, KAIYUAN-2B exhibits compelling data and compute efficiency. By training on only 2.2T tokens, KAIYUAN-2B matches or exceeds results from counterparts that utilize significantly larger data budgets, such as SmolLM2-1.7B. While a performance gap remains relative to the Qwen series, likely due to their 36T-token scale and proprietary data mixtures, KAIYUAN-2B establishes a fully open, parameter-efficient, and compute-economic alternative for the research community.

8 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 Selective Repetition, and Comprehensive 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 data preprocessing framework, and the final pretraining dataset, we provide a complete, transparent recipe for high-quality LLM pretraining. We believe KAIYUAN-2B will facilitate further exploration in the open-source LLM ecosystem, pushing the frontier of what is achievable under limited resources.

9 Limitations

Primarily, although KAIYUAN-2B has made progress, there is still a great gap between our model and the leading open-weight model, like Qwen3 series (Yang et al., 2025). Moreover, due to the scale and cost of our pretraining setup, we may not conduct extensive ablation studies on each design choice. For example, how the multi-phase design is compared to the conventional two-phase mid-training strategy, and how the multi-dataset curriculum is effective over the data mixture in the last three phases. And there is still space for more quantitative design, like how to decide the repetition number and sampling ratio for each specific dataset, which can be a promising future direction.

AI Assistant Usage. We use the AI Assistant to polish the paper writing.

References

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, and 3 others. 2025a. Smollm2: When smol goes big - data-centric training of a small language model. *CoRR*, abs/2502.02737.

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, and 3 others. 2025b. Smollm2: When smol goes big – data-centric training of a small language model. *Preprint*, arXiv:2502.02737.

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

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

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.

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. *Preprint*, arXiv:2310.10631.

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, and 4 others. 2025. SmolLM3: smol, multilingual, long-context reasoner. https://huggingface.co/blog/smollm3.

Aarti Basant, Abhijit Khairnar, Abhijit Paithankar, Abhinav Khattar, Adithya Renduchintala, Aditya Malte, Akhiad Bercovich, Akshay Hazare, Alejandra Rico, Aleksander Ficek, Alex Kondratenko, Alex Shaposhnikov, Alexander Bukharin, Ali Taghibakhshi, Amelia Barton, Ameya Sunil Mahabaleshwarkar, Amy Shen, Andrew Tao, Ann Guan, and 80 others. 2025. NVIDIA nemotron nano 2: An accurate and efficient hybrid mamba-transformer reasoning model. *CoRR*, abs/2508.14444.

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.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. 2024. Smollm-corpus.

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*, pages 7432–7439. AAAI Press.

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, and 39 others. 2021. Evaluating large language models trained on code. *CoRR*, abs/2107.03374.

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)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

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.

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

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].

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

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

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*, ICML'17, page 933–941. JMLR.org.

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, and 80 others. 2024. DeepSeek-V3 technical report. *CoRR*, abs/2412.19437.

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*, pages 19822–19835.

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. *Preprint*, arXiv:2505.02881.

Gemini Team. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *CoRR*, abs/2507.06261.

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, and 24 others. 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)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.

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*, pages 5005–5033. Association for Computational Linguistics.

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*.

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. *Preprint*, arXiv:2308.10755.

Conghui He, Wei Li, Zhenjiang Jin, Chao Xu, Bin Wang, and Dahua Lin. 2024. Opendatalab: Empowering general artificial intelligence with open datasets. *Preprint*, arXiv:2407.13773.

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.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021b. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and

Jacob Steinhardt. 2021c. 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*.

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*, volume EMNLP 2020 of *Findings of ACL*, pages 4246–4253. Association for Computational Linguistics.

Yiwen Hu, Huatong Song, Jia Deng, Jiapeng Wang, Jie Chen, Kun Zhou, Yutao Zhu, Jinhao Jiang, Zican Dong, Wayne Xin Zhao, and 1 others. 2024. Yulan-mini: An open data-efficient language model. *arXiv preprint arXiv:2412.17743*.

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*.

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

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*.

Oleksii Kuchaiev, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Kriman, Stanislav Beliaev, Vitaly Lavrukhin, Jack Cook, Patrice Castonguay, Mariya Popova, Jocelyn Huang, and Jonathan M. Cohen. 2019. Nemo: a toolkit for building AI applications using neural modules. *CoRR*, abs/1909.09577.

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

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, and 4 others. 2024. Tülu 3: Pushing frontiers in open language model post-training.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024a. 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*, pages 11260–11285. Association for Computational Linguistics.

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, and 40 others. 2024b. Datacomp-lm: In search of the next generation of training sets for language models. *Preprint*, arXiv:2406.11794.

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, and 40 others. 2024c. 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*.

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, and 7 others. 2025. Model merging in pre-training of large language models. *CoRR*, abs/2505.12082.

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. *Preprint*, arXiv:2301.13688.

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, and 47 others. 2024. Starcoder 2 and the stack v2: The next generation. *Preprint*, arXiv:2402.19173.

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. *Preprint*, arXiv:2511.18903.

Meta AI. 2024. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

David Mizrahi, Anders Boesen Lindbo Larsen, Jesse Allardice, Suzie Petryk, Yuri Gorokhov, Jeffrey Li, Alex Fang, Josh Gardner, Tom Gunter, and Afshin Dehghan. 2025. Language models improve when pretraining data matches target tasks. *CoRR*, abs/2507.12466.

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*.

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, and 2 others. 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.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *Preprint*, arXiv:2306.02707.

NVIDIA, :, Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, Akhiad Bercovich, Aleksander Ficek, Aleksandr Shaposhnikov, Alex Kondratenko, Alexander Bukharin, Alexandre Milesi, Ali Taghibakhshi, Alisa Liu, Amelia Barton, and 340 others. 2025. Nvidia nemotron 3: Efficient and open intelligence. *CoRR*.

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].

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].

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. 2023. Openwebmath: An open dataset of high-quality mathematical web text. *Preprint*, arXiv:2310.06786.

Guilherme Penedo. 2025. 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.

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*.

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, and 80 others. 2024. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118.

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*, pages 8732–8740. AAAI Press.

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)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

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

KaShun SHUM, Yuzhen Huang, Hongjian Zou, dingqi, YiXuan Liao, Xiaoxin Chen, Qian Liu, and Junxian He. 2025. Predictive data selection: The data that predicts is the data that teaches. In *Forty-second International Conference on Machine Learning*.

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

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Raghavi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, and 17 others. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15725–15788. Association for Computational Linguistics.

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*, pages 2459–2475. Association for Computational Linguistics.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C).

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)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari Morcos. 2023. D4: improving LLM pretraining via document de-duplication and diversification. 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*.

Evan 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, Allyson Ettinger, and 23 others. 2025. 2 OLMo 2 furious (COLM's version). In *Second Conference on Language Modeling*.

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. *Preprint*, arXiv:2410.18505.

Xingjin Wang, Howe Tissue, Lu Wang, Linjing Li, and Daniel Dajun Zeng. 2025. Learning dynamics in continual pre-training for large language models. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.

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, and 11 others. 2023. Skywork: A more open bilingual foundation model. *Preprint*, arXiv:2310.19341.

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*.

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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 40 others. 2025. Qwen3 technical report. *CoRR*, abs/2505.09388.

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, and 43 others. 2024a. Qwen2 technical report. *CoRR*, abs/2407.10671.

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, and 22 others. 2024b. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

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)*, pages 5374–5400, Vienna, Austria. Association for Computational Linguistics.

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):872–885.

Yijiong Yu, Ziyun Dai, Zekun Wang, Wei Wang, Ran Chen, and Ji Pei. 2025a. Opencsg chinese corpus: A series of high-quality chinese datasets for LLM training. *CoRR*, abs/2501.08197.

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

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*, pages 15–28. USENIX Association.

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

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*, pages 12360–12371.

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*, pages 4168–4189, Vienna, Austria. Association for Computational Linguistics.

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. 2021a. Cpm-2: Large-scale cost-effective pre-trained language models. *AI Open*, 2:216–224.

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, and 6 others. 2021b. Cpm: A large-scale generative chinese pre-trained language model. *AI Open*, 2:93–99.

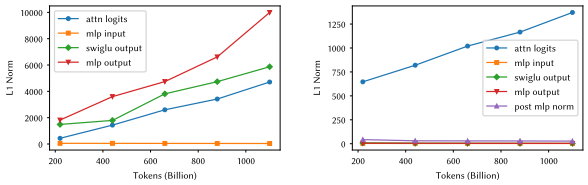
Fan Zhou, Zengzhi Wang, Nikhil Ranjan, Zhoujun Cheng, Liping Tang, Guowei He, Zhengzhong Liu, and Eric P. Xing. 2025. Megamath: Pushing the limits of open math corpora. *CoRR*, abs/2504.02807.

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.

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, and 19 others. 2024. Yulan: An open-source large language model. *CoRR*, abs/2406.19853.

Appendices

A Architecture Design and Training Stability



(a) Activation statistics of the baseline architecture

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

Figure 5: Comparison of internal activation magnitudes before and after architectural optimization. The experiment is conducted with a 3B model.

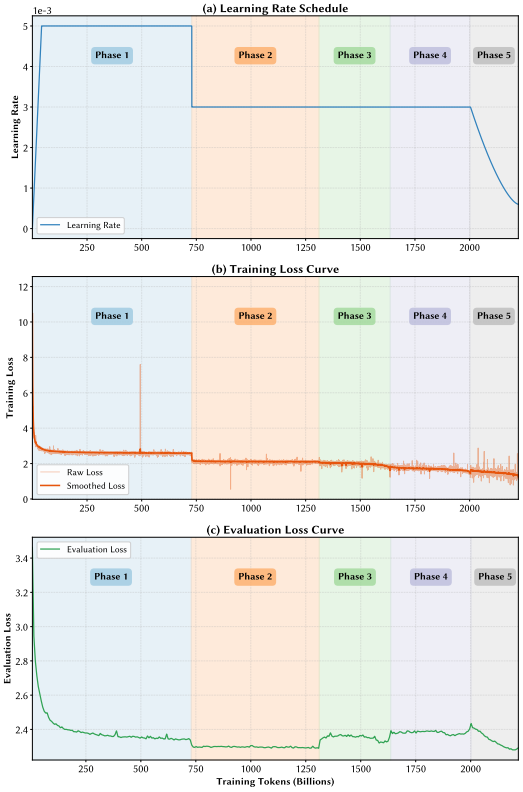


Figure 6: Learning Rate Schedule, Training Loss, and Validation Loss Curves across five phases.

KAIYUAN-2B is trained on Huawei Ascend 910A accelerators, which are similar to NVIDIA V100s in supporting only FP16 precision. 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 (Dauphin et al., 2017), RMSNorm (Zhang and Sennrich, 2019), and RoPE (Su et al., 2024). 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 5a, 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 (Bello et al., 2017) and Sandwich Normalization (Ding et al., 2021). This follows the design choices of Gemma 2 (Rivière et al., 2024). These techniques place strict bounds on activation values. As shown in Figure 5b, 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 (Henry et al., 2020). The full configuration of KAIYUAN-2B is listed in Table 3 and the implementation details are discussed in Section E.1.

Loss Dynamics and Transition Analysis. The training trajectories are visualized in Figure 6. Training loss exhibits sharp drops at phase transitions, driven by the introduction of a higher ratio of low-perplexity mathematical and code data. Alongside the quality-based curriculum, which accelerates convergence toward the end of each phase, we observe a different loss curve pattern from traditional scaling law (Kaplan et al., 2020). Moreover,

as the validation set consists of the high-quality DCLM subset, the validation loss shows anomalous increases during later phases. This "misalignment" reflects the model's specialization: as KAIYUAN-2B focuses on the specialized data (math and code), its performance on general English validation sets slightly degrades, highlighting the trade-offs inherent in multi-domain curriculum progression. This observation aligns with the prior work on continual training scaling laws (Wang et al., 2025).

B Quality-Score Quantile Benchmarking

We show full quantile benchmarking results in Figures 7 and 8. The overall observations are discussed in Section 3 in detail. The DCLM-Baseline leading experiments are shown in Figure 7 and Fineweb-Edu leading experiments are shown in Figure 8.

C Datasets Used in Training

Table 18 is a comprehensive list of all datasets used in the training process of KAIYUAN-2B. All datasets are publicly available to acquire, and most of them are hosted on Hugging Face unless otherwise noted.

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 18, so that they 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 can use these same datasets to reproduce our results and further adapt and/or publish both the modified datasets and models at will, free from potential 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* (NVIDIA, 2025) 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 Figures 9 to 13, respectively. Moreover, we list the detailed dataset composition for each phase in Tables 13 to 17, 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 13) and Fineweb-Edu-EN in Phase 2 (Table 14).

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 13), 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 13).

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 15), 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 17).

In addition, LLM360-Math is a deduplicated subset of the MegaMath dataset (Zhou et al., 2025), and we select only the top 5% of rows from the English partition of the FinePDFs dataset (Kydlíček et al., 2025), according to Fineweb-Edu classifier scores (Penedo et al., 2024).

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

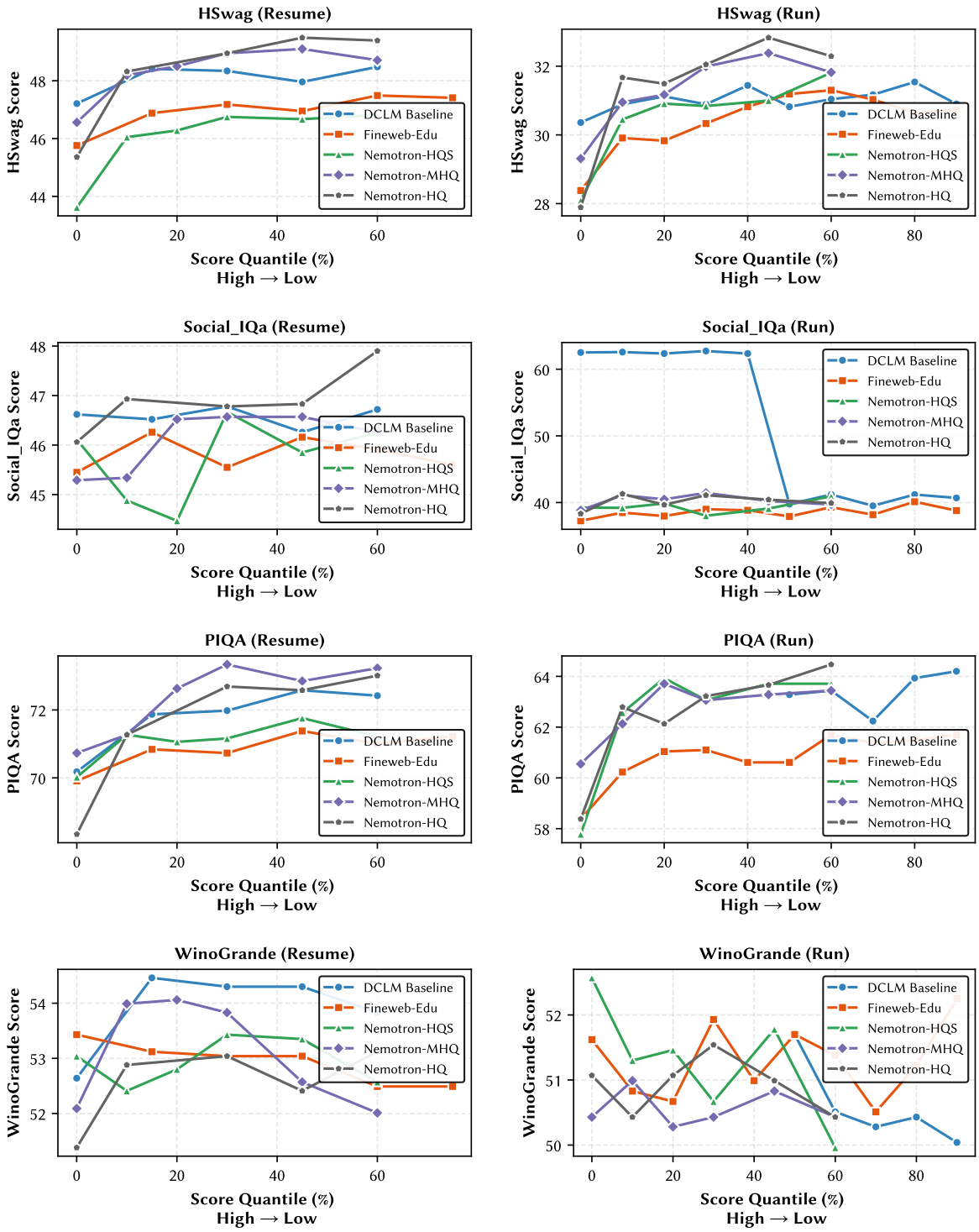


Figure 7: Quantile Benchmarks: DCLM-Baseline is better on understanding-oriented benchmarks.

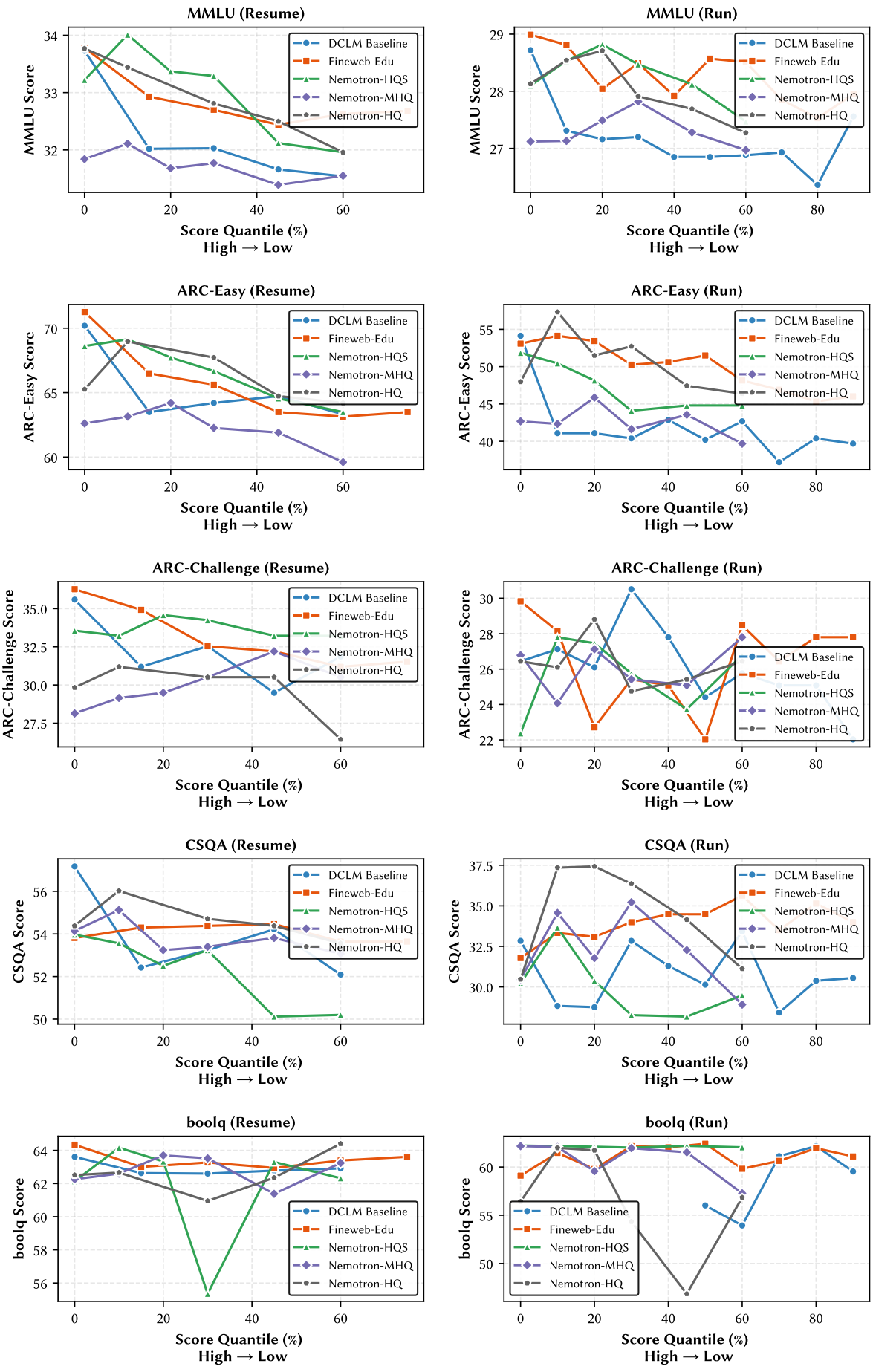


Figure 8: Quantile Benchmarks: FineWeb-Edu is better on knowledge-oriented benchmarks.

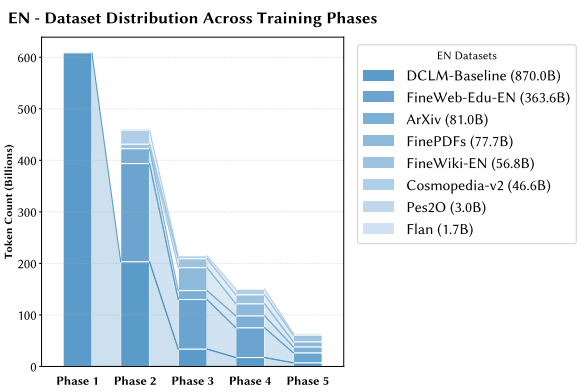


Figure 9: Phase-wise Dataset Mixture: English.

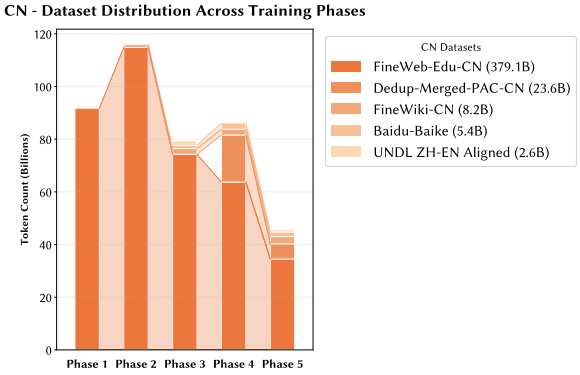


Figure 10: Phase-wise Dataset Mixture: Chinese.

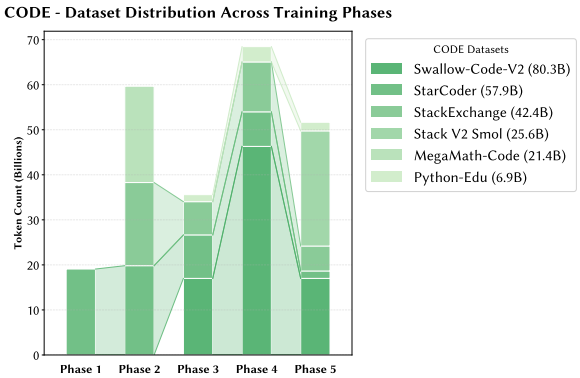


Figure 11: Phase-wise Dataset Mixture: Code.

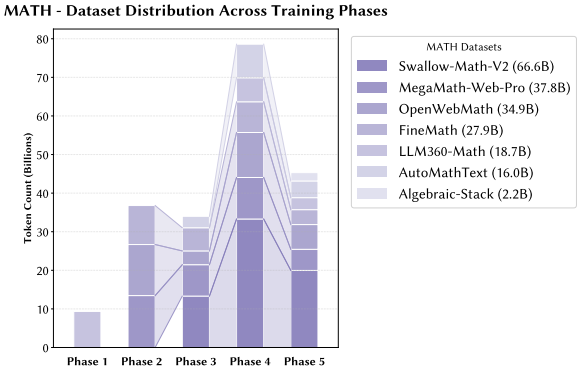


Figure 12: Phase-wise Dataset Mixture: Math.

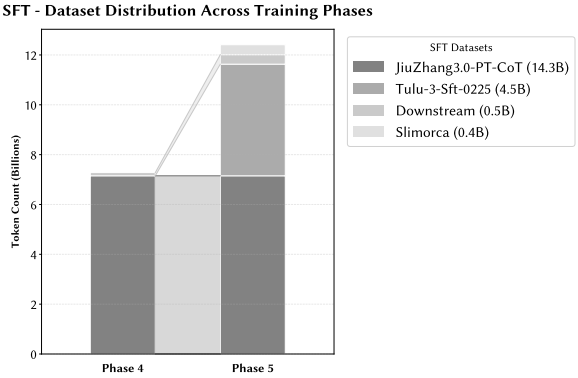


Figure 13: Phase-wise Dataset Mixture: SFT.

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 3, we present the details of our training hyperparameter configuration in three parts:

- For the model architecture, we primarily follow Qwen3-1.7B (Yang et al., 2025) and adopt the vocabulary from the Qwen series (Yang et al., 2024a, 2025). We use $\theta = 10000$ for RoPE (Su et al., 2024) to support a context length of 4K. The Soft-Capping threshold is set to 30.0, as discussed in Section A.
- 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.

Table 3: Training Hyperparameter Configuration.

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

E.3 Model Average

Following recent work (Luo et al., 2025), 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 4. 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 (Li et al., 2025), 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 roughly 10B tokens, which are then used for the respective training scenarios.

Training from Scratch. In the training-from-scratch scenario, we train a model with the Qwen3-0.6B architecture (Yang et al., 2025). Following the default configuration in Table 3, we conduct a small-scale experiment using the settings detailed in Table 5, 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 (Yang et al., 2024b). We then train over approximately 8.4B tokens from the quantile data chunks using the configuration specified in Table 6. 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 7 and 8, 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 4: Model Performance Across Checkpoints.

Ckpt 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

Compute Cost Comparison. We use the DCLM-Baseline as a reference for comparison, as discussed in Section 3. Following previous work (Kaplan et al., 2020), we estimate the compute budget using the formula $C = 6ND$, where N represents the number of model parameters and D represents the data size. Consequently, the compute cost for the 0.6B model trained on 42B tokens is approximately $(0.6 \times 42)/(2 \times 609) \approx 2.07\%$ relative to the baseline. Similarly, we conclude that this accounts for less than 0.6% of the total pretraining budget.

Table 5: 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 6: 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 (Luo et al., 2025). 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 (Hägele et al., 2024; Tian et al., 2025), decaying to a near-zero final learning rate. The detailed experimental configuration is provided in Table 7.

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[8] (Luo et al., 2025), which integrates curriculum learning with either no or moderate LR decay, accompanied by model averaging over the final checkpoints.

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

[8]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 7: 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 8: 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

decay factor), replicating the methodology from CMA (Luo et al., 2025).

Evaluation Settings. In Table 1, we evaluate performance on a high–signal-to–noise-ratio benchmark subset (*Core* in Table 1) comprising MMLU (Hendrycks et al., 2021b), ARC (Clark et al., 2018), and CSQA (Talmor et al., 2019), following established practices in prior work (Heineman et al., 2025; Luo et al., 2025). These benchmarks provide strong discriminative power for identifying performance differences between training approaches.

E.6 Data Processing Framework

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

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.

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.

High Performance: To handle hundreds of terabytes of data, the framework must be optimized to reduce computation overhead.

To meet these demands, we developed **KAIYUAN-SPARK**, a distributed data processing framework built on Spark (Zaharia et al., 2012). KAIYUAN-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 KAIYUAN-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 KAIYUAN-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 (Yu et al., 2021) 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.

F Model Performance across Benchmarks (Full Table)

Table 2 merges the results from both Tables 10 and 11, 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 14. These plots show that KAIYUAN-2B lies on the frontier of fully open-source models.

Table 9: Model Parameter Statistics Comparison.

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

Table 10: Performance across Chinese, Mathematics, and Coding benchmarks.

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								
Kaiyuan-2B	2B	46.30	49.25	51.33	30.34	56.42	42.68	46.05

* Cited from official reports or original papers.

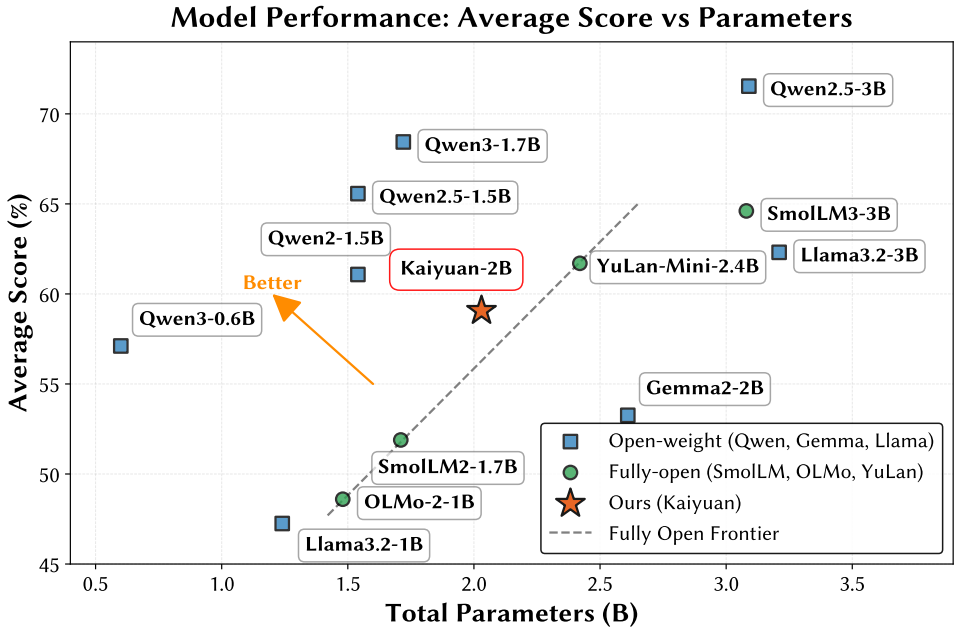


Figure 14: **Performance over total parameters.** KAIYUAN-2B still surpasses the frontier of fully open-source models at a similar scale and narrows the gap to leading open-weight models like Qwen2-1.5B. See Table 2 for full results.

Table 11: 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											
Kaiyuan-2B	2B	53.90	66.10	82.89	78.53	67.40	58.13	74.37	62.59	65.75	67.74

Table 12: 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																	
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

Table 13: 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%

Table 14: 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 15: 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 16: 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 17: 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%

Table 18: All Datasets Used in the Training of KAIYUAN-2B.

Name	Type	Hugging Face ID	#Tokens[0]	License(s)
DCLM-Baseline	English	mlfoundations/dclm-baseline-1.0 (Li et al., 2024b)	4T	CC BY 4.0[1]
FineWiki-EN	English	HuggingFaceFW/finewiki (Penedo, 2025)	8.7B	CC BY-SA 4.0[6]
FinePDFs	English	HuggingFaceFW/finepdfs (Kydlíček et al., 2025)	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 (Ben Allal et al., 2024)	220B	ODC-By 1.0[1]
ArXiv	English	togethercomputer/RedPajama-Data-1T (Computer, 2023)	28B	Metadata: CC0 1.0 (arXiv info, 2025b) Content: various (arXiv info, 2025a)
Cosmopedia-v2	English	HuggingFaceTB/smollm-corpus (Ben Allal et al., 2024)	27B	ODC-By 1.0
FineWiki-CN	Chinese	HuggingFaceFW/finewiki (Penedo, 2025)	1.1B	CC BY-SA 4.0[6]
Fineweb-Edu-CN	Chinese	opencsg/Fineweb-Edu-Chinese-V2.1 (Yu et al., 2025b)	1.5T	OpenCSG Community License (Community, 2024), 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
Dedup-Merged-PAC-CN[4]	Chinese	BAAI/CCI-Data BAAI/CCI2-Data BAAI/CCI3-Data (Wang et al., 2024) Skywork/SkyPile-150B (Wei et al., 2023) OpenDataLab/WanJuan1.0 (He et al., 2023, 2024)[5] BAAI/IndustryCorpus BAAI/IndustryCorpus2 (Shi et al., 2024) WuDaoCorpus2.0 (Zhang et al., 2021a,b)[5]	178B	CCI{,2,3}-Data: CCI Usage Agreement (of Artificial Intelligence, 2023) SkyPile-150B: Skywork Community License (Skywork-AI, 2023), 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 (Paster et al., 2023)	14.7B	ODC-By 1.0[1]
FineMath	Math	HuggingFaceTB/finemath (Allal et al., 2025b)	10B	ODC-By 1.0
MegaMath-Web-Pro	Math	LLM360/MegaMath (Zhou et al., 2025)	300B	ODC-By 1.0
AutoMathText	Math	math-ai/AutoMathText (Zhang et al., 2025)	8.7B	CC BY-SA 4.0
SwallowMath-v2	Math	tokyotech-llm/swallow-math-v2 (Fujii et al., 2025)	32B	Apache 2.0
StarCoder	Code	bigcode/starcoderdata (Kocetkov et al., 2022)	250B	Original Licenses[2]
Stack V2 Smol	Code	bigcode/the-stack-v2 (Lozhkov et al., 2024)	900B	Original Licenses[2]
StackExchange	Code	togethercomputer/RedPajama-Data-1T (Computer, 2023)	20B	CC BY-SA 2.5/3.0/4.0[3]

Continued on next page

Table 18: All Datasets Used in the Training of KAIYUAN-2B. (Continued)

Name	Type	Hugging Face ID	#Tokens[0]	License(s)
Python-Edu	Code	HuggingFaceTB/smollm-corpus (Lozhkov et al., 2024; Ben Allal et al., 2024)	3.4B	ODC-By 1.0, Original Licenses[2]
Algebraic-Stack	Code	typeof/algebraic-stack (Azerbayev et al., 2023; Paster et al., 2023)	11B	ODC-By 1.0[1]
Swallow-Code-v2	Code	tokyotech-llm/swallow-code-v2 (Fujii et al., 2025)	49.8B	Apache 2.0
SlimOrca	SFT	Open-Orca/SlimOrca (Mukherjee et al., 2023; Longpre et al., 2023)	190M	MIT
JiuZhang3.0-Corpus-CoT	SFT	ToheartZhang/JiuZhang3.0-Corpus-CoT (Zhou et al., 2024)	358B	*Not Specified*
Tulu-3-Sft-0225	SFT	allenai/tulu-3-sft-mixture (Lambert et al., 2024)	640M	ODC-By 1.0 (mixed)
downstream[4]	SFT	cais/mmlu (Hendrycks et al., 2021b,a) openai/gsm8k (Cobbe et al., 2021) allenai/ai2_arc (Clark et al., 2018) allenai/openbookqa (Mihaylov et al., 2018) Rowan/hellaswag (Zellers et al., 2019) allenai/winogrande (Sakaguchi et al., 2020)	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 (Common Crawl, 2024).

[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.