Kaiyuan-2B: Open LLM Pretraining Guided by Quantile Benchmarking of Heterogeneous Corpora

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

Abstract

Fully open pretraining now has increasingly reproducible recipes and accessible web-scale corpora. What remains missing is a principled way to merge scored open-web data. Corpora such as DCLM Baseline and FineWeb-Edu are built with different filters and scorers, so their raw scores are not directly comparable. Because web data is large, heterogeneous, and token-budget dominant, choosing an additional retained fraction is a consequential compute decision rather than a minor sampling detail. We formulate this setting as *heterogeneous top-p mixing*: choosing top-p retained fractions for each corpus along its score distribution under a fixed total token budget. We propose *quantile benchmarking*, a proxy-training framework that probes fixed score quantiles from each source and compares them through a shared downstream-utility metric. The resulting measurements reveal benchmark-dependent source orderings, large within-source utility variance, and a strong knowledge bias in current scorers. In particular, higher-score regions often degrade broader commonsense benchmarks such as PIQA and HellaSwag. We then fit benchmark-conditioned utility curves and derive top-p mixing cutoffs through a global utility threshold. 40B-token ablation experiments show that quantile-benchmarking-guided thresholds improve over both uniform mixing and heuristic top-p filtering. Finally, we apply the resulting recipe to train a 2B model over 2.2T tokens with a fully open recipe, reaching the Pareto frontier of parameter efficiency among compared fully open models of similar scale.

1 Introduction

Fully open language model pretraining has made recipes increasingly reproducible, with recent releases documenting architectures, hyperparameters, final data mixtures, ablation protocols, and scaling behavior in increasing detail [3, 8, 9, 34, 70]. But data recipes are harder to reuse than optimizer or architecture choices, because they depend on the training objective, available corpora, licensing constraints, and token budget. Before training, practitioners still need a principled way to decide which open data sources to collect, how deeply to filter each source, and how to combine the retained subsets.

Open curated web corpora are central to this decision. Red Pajama [21], RefinedWeb [58], DCLM Baseline [41], FineWeb-Edu [59], and Nemotron-CC-v2 [66] have become common infrastructure for fully open pretraining, data-mixing studies, and recipe ablations [47, 70, 77, 78]. These collections are processed, and often scored or filtered, produced by large-scale curation pipelines. Compared to scarce high-value sources such as math, code, and reasoning data, curated web corpora are massive,

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

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

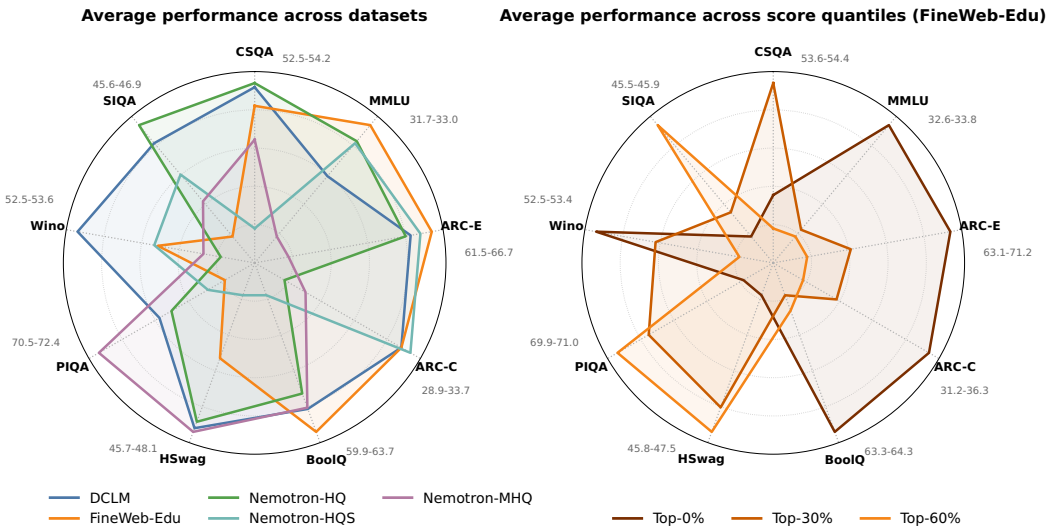

Figure 1: **Capability profiles of heterogeneous web corpora.** Left: average capability profiles of five web sources in the resume setting, averaged over the shared score quantiles 0, 30, and 60, where smaller quantile indices correspond to higher-scored windows. Right: capability profiles of FineWeb-Edu at the same three score quantiles. In both panels, each benchmark axis is normalized independently. The figure shows two effects: source ordering is benchmark-dependent, and FineWeb-Edu's higher-scored windows help MMLU and ARC-E/C but can reduce PIQA and HellaSwag utility. Full curves are provided in Figures 6 and 7.

noisy, internally diverse, and thus token-budget dominant in pretraining. A small change in their retained fraction can therefore differ by tens or hundreds of billions of training tokens.

The open question is how these retained fractions should be chosen across scored corpora. A natural recipe would combine their strengths by retaining high-scoring regions of each source under a fixed token budget. This is not standard source-weight optimization. We call the setting *heterogeneous top-p mixing*: choose source-specific retained top-p fractions for several scored corpora, then merge the retained subsets into the final mixture. Filtering and mixing are coupled because increasing one source's retained fraction both changes its token share and moves selection into lower-scored regions, so the average quality of the retained subset can decrease.

The difficulty is that the rankings are dataset-specific, and there is no common score scale across sources. For example, DCLM Baseline and FineWeb-Edu both derive from web-scale text, but they are built with different filtering pipelines, different scorers, and different score semantics. DCLM Baseline uses ELI5 and OH-2.5 as reference datasets to train a FastText scorer, while FineWeb-Edu uses LLM judgments to train an educational-value scorer [41, 59]. Therefore, a high score in one source does not mean the same as a high score in another. Figure 5 shows this mismatch on a shared document sample: sorting by one scorer produces little structure in the other scorer. As a result, a shared raw-score threshold cannot reliably determine how to combine these corpora. We therefore need a shared downstream utility metric for comparing source-specific score regions.

Quantile benchmarking. We propose *quantile benchmarking* to make this problem measurable. For each source, we probe several score quantiles using fixed-size training chunks and evaluate the resulting proxy checkpoints on downstream benchmarks. These benchmark scores provide a shared utility metric for comparing sources whose raw scores are not calibrated to one another. Compared to prior work, which typically uses top-p filtering to evaluate a scorer [41, 51], quantile benchmarking captures fine-grained information on how different regions of the score distribution contribute to different capabilities. It serves both as a dataset benchmark and as a calibration tool for corpus-provided scores. Figure 1 provides representative dataset profile comparisons.

From observations to mixture deduction and data-centric strategies. Quantile benchmarking supports both diagnosis and recipe design. It explicitly reveals benchmark-dependent source ordering, large within-source utility variance, and knowledge-biased scorer behavior, as previewed in Figure 1. In particular, it is surprising that common higher-score regions can flatten or degrade broader commonsense benchmarks such as PIQA and HellaSwag. Moreover, the same probes can be converted into a practical mixture. We fit benchmark-conditioned utility functions and solve for a

global utility threshold. The threshold then induces retained fractions for different sources. The measurements also connect to prior work on data-centric strategies, including curriculum model averaging [47] and selective repetition [69].

Validation and open pretraining. We validate the framework in controlled 40B-token ablations. Moving from uniform mixing to heuristic top-p filtering improves the Core benchmark suite. Replacing the heuristic thresholds with quantile-benchmarking-guided thresholds gives an additional gain, and a quality curriculum adds a further improvement by leveraging utility variance. These experiments provide the main controlled evidence for the proposed mixture deduction. We then apply the resulting recipe components (top-p mixing, curriculum averaging, and selective repetition) to a fully open 2B-scale pretraining run, called KAIYUAN-2B. It serves as a practical end-to-end transfer case and exhibits competitive pretraining performance compared to similar scale models (Figure 2).

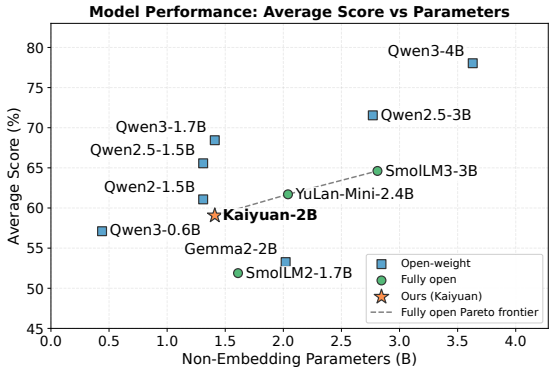

Figure 2: **Fully open 2B release case.** KAIYUAN-2B lies on the parameter-efficiency frontier of compared fully open models with complete fifteen-task results (Section J).

Our contributions are threefold.

(1) We formulate open-web recipe design as a *heterogeneous top-p mixing* problem, and propose *quantile benchmarking* to calibrate source-specific scores through downstream utility.

(2) We quantify benchmark-dependent source ordering, scorer bias, and within-source utility variance in widely used open curated web corpora.

(3) We validate benchmarking-guided mixing and related data-centric strategies through controlled 40B-token ablations, and apply the resulting recipe in a fully open 2.2T-token, 2B-scale pretraining run. We release all assets (including model weights, data, and code) under the Apache 2.0 license.[3]

2 Related Work

Data recipe in fully open pretraining. Fully open language model pretraining releases not only model weights, but also data recipes, hyperparameter settings, and training details [3, 8, 9, 24, 34, 54, 70, 85, 95]. This transparency supports reproduction, but final data mixtures alone do not explain how those mixtures were chosen. The gap matters because available corpora, licensing constraints, target capabilities, and token budgets can all change across reproduction settings. Our work focuses on this design step, especially mixture design over dominant open-web corpora.

Data mixing with uniform sampling. Our work is related to data mixture optimization. Methods such as DoReMi, RegMix, data mixing laws, MDE, MergeMix, and TiKMiX estimate how mixture weights affect downstream or language-modeling performance and search for better source ratios [10, 43, 71, 74, 79, 84]. These methods are highly relevant, but they usually treat each component domain or dataset as a fixed unit and optimize how much to sample from it. In contrast, our top-p mixing formulation chooses how deeply to retain each scored corpus. The retained fraction determines both the amount of data contributed by that source and the average quality and capability profile of the retained subset.

Score and dataset bias. Score and dataset bias have been observed in prior work. For example, Penedo et al. [59] shows that educational filtering changes content coverage, including stronger coverage of Wiki-like content. Wettig et al. [78] studies domain organization for processed web data, and Su et al. [66] diagnoses partial orthogonality between DCLM and FineWeb-Edu scores. These

[3] Model weights and release documentation: `https://huggingface.co/thu-pacman/PCMind-2.1-Kaiyuan-2B`. Dataset, preprocessing, and training-code links are listed in Section I.

results show that different web corpora and scorers capture different quality signals. Our contribution is a quantile-resolved diagnosis: we measure how score regions within each source contribute to downstream tasks, and show that higher-score samples can improve knowledge-heavy benchmarks while hurting or flattening broader commonsense benchmarks.

Additional discussion appears in Section B.

3 Problem Setup: Heterogeneous Top-p Mixing

We consider a set of scored web corpora $\mathcal{D}_1, \ldots, \mathcal{D}_m$. Each source $\mathcal{D}_i$ has its own local score function $s_i(x)$ and total available token volume N_i. Here, x denotes a document or training sequence from $\mathcal{D}_i$. These raw scores are not aligned across sources. They come from different filters and different scorers, so they do not share a common numerical meaning.

In conventional domain mixing, one chooses source weights w_i and then samples uniformly within each source. This implicitly assumes that each source is internally homogeneous enough for its score structure to be ignored. This assumption fails for scored web corpora, as shown in Section 5. The top 10% of one source can be very different from a uniform sample of the same source, and ignoring this difference wastes information produced by the curation pipeline.

We instead define the recipe by source-specific retention quantiles $q_i \in (0, 100]$. For each source, we select a retention set $\mathcal{R}_i(q_i) = \{x \in \mathcal{D}_i : x \text{ belongs to the top-}q_i\% \text{ under } s_i\}$. We use *retained fraction* for the numeric value q_i, *top-p filtering* for the single-source selection action, and *top-p mixing* for the multi-source recipe formed after these selections. The final mixture is the union of retained token pools, without an additional source-level resampling weight: $\mathcal{M}(q_1, \ldots, q_m) = \bigcup_i \mathcal{R}_i(q_i)$. The effective source weight is therefore proportional to the retained token volume, $w_i(q_i) = \frac{N_i q_i}{\sum_\ell N_\ell q_\ell}$. This makes filtering and mixing coupled. Changing q_i changes both the internal quality profile of source i and its share in the final token budget. Increasing q_i also moves selection deeper into lower-scored regions, lowering the average score of the retained subset by construction. Top-p mixing is therefore non-trivial even before raw-score incompatibility is considered. It needs a shared utility metric that can align heterogeneous source scores; we state the practical assumptions in Section F. Quantile benchmarking provides this metric through fine-grained measurements of source-specific score regions.

4 Quantile Benchmarking

Quantile benchmarking pipeline. We propose quantile benchmarking to estimate how score quantiles within a source map to downstream capability. For a chosen source, we first select a set of target quantiles $Q = \{q_1, \ldots, q_K\}$. Around each quantile, we construct a fixed-size probing chunk and train a proxy model on that chunk. We then evaluate the resulting checkpoint on a target benchmark suite. Repeating this process yields discrete observations $\{(q_j, y_j)\}$ that describe how downstream utility changes over the source's score distribution. These observations serve two purposes. They reveal how a scorer behaves across different benchmarks, and they identify benchmark dimensions that can serve as shared metrics for top-p mixing. The method assumes that the main utility trend can be approximated from a small number of quantile probes. The resulting trends are examined in Section 5; resolution limits are discussed below and in Section C.1.

In our implementation, a probing chunk is defined by a target score quantile and a fixed token budget. We first sort each source by its local quality score. Starting from the target quantile boundary, we then collect the next fixed-size block of tokens in score order, from high to low. This produces a quantile-anchored and token-controlled window in the source-specific score distribution. Additional setup details are summarized in Section C.1; we discuss the resulting quantile resolution limitation, probing overhead, and cost estimation in Section C.1. For DCLM Baseline and FineWeb-Edu, quantiles are computed after global fuzzy deduplication; Nemotron-CC-v2 has a lower duplication rate, so we do not apply an additional deduplication step (Section C.1).

We run quantile benchmarking in two regimes. In the *from-scratch* regime, the proxy model is trained only on the probing chunk. This estimates the chunk's raw utility. In the *resume* regime, the proxy model starts from a pretrained checkpoint and continues on the probing chunk. This estimates its refinement utility. The two regimes provide different views of the same source and help

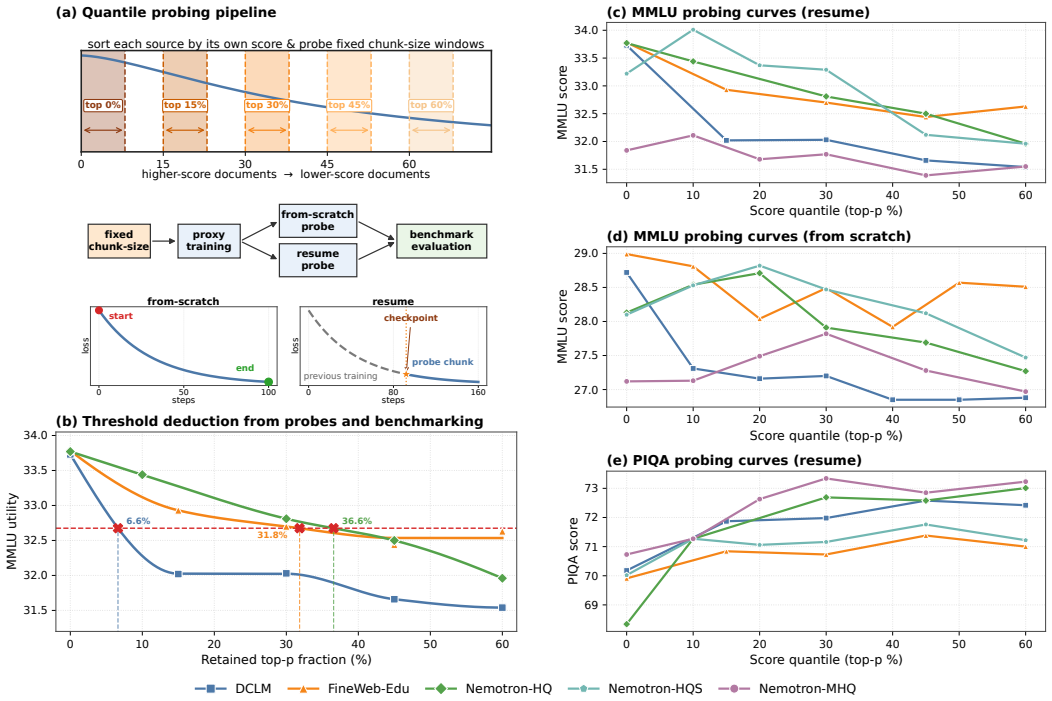

Figure 3: **Quantile benchmarking and threshold deduction.** Panel (a) shows the quantile probing pipeline: sort each source by its own score, collect fixed-token chunks anchored at selected top-p boundaries, train proxy checkpoints, and evaluate them on downstream benchmarks. Panel (b) converts MMLU utility observations into fitted within-source quantile cutoffs, not final mixture shares, under a 40B-token budget by fitting monotone utility curves and solving a global threshold. Panels (c)–(e) show representative probing curves for MMLU and PIQA, restricted to the top-60% region because the induced mixtures mainly operate in this high-score range. Across different scoring, high-score samples are beneficial to the MMLU, but hurt PIQA. Resume and from-scratch experiments show overall aligned trends for MMLU. Section 5 shows finding details.

us study whether the observed ordering is robust to training context. The whole pipeline of quantile benchmarking and representative benchmarking results can refer to Figure 3.

Difference from top-p filtering evaluation. This procedure differs from standard top-p filtering evaluation. A single top-p experiment measures only the cumulative average quality above one threshold. It cannot show whether utility changes smoothly, reverses direction, or varies sharply inside a high-score region. It also hides variance within a source because all retained samples are averaged into one training run. Quantile benchmarking gives a more fine-grained view: each probe measures a local region of the score distribution. The right column of Figure 3 illustrates why these additional probes are informative: they surface benchmark-dependent utility trends that single-threshold filtering would hide.

5 What Quantile Benchmarking Reveals about Open Web Data

We apply quantile benchmarking to three representative English web sources: DCLM Baseline [41], FineWeb-Edu [59], and Nemotron-CC-v2 [9, 66]. For Nemotron-CC-v2, which does not expose per-sample quality scores, we annotate the data with PreSelect scores [64]. We also separately study its high-quality (HQ), high-quality synthetic (HQS), and medium-high-quality (MHQ) partitions. We focus on these sources for two reasons. First, they are common open-web corpora in fully open pretraining, recipe ablations, and data-mixing studies. Second, they represent different curation styles. DCLM uses score-based filtering with a lightweight learned scorer. FineWeb-Edu uses educational-value refinement with an LLM-involved annotation pipeline. Nemotron uses a heavier refinement pipeline with released stratified partitions and additional synthetic data. Quantile benchmarking therefore tests both the corpora and the assumptions behind their scoring or refinement pipelines.

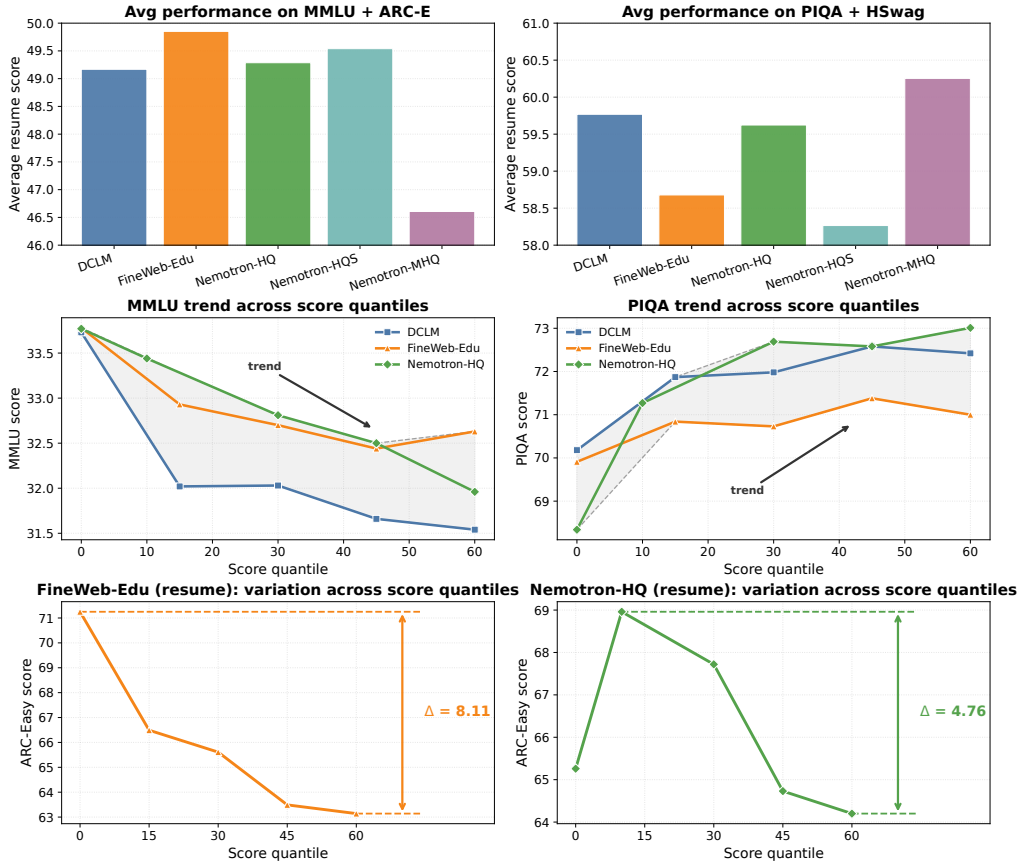

Figure 4: **Quantile benchmarking reveals benchmark-conditioned source orderings and large within-source utility variation.** Top row: average resume performance by source on two benchmark groups. We average over the shared score quantiles 0, 30, and 60. The left panel averages MMLU and ARC-Easy, while the right panel averages PIQA and HellaSwag. Middle row: score-quantile trends differ by benchmark within the top-60% region; the gray band connects the upper and lower source envelopes. The arrows mark the dominant direction. Bottom row: FineWeb-Edu and Nemotron-HQ both show large ARC-Easy variation across score quantiles in the resume regime. Dashed guides mark the best and worst probed quantiles.

Finding 1: Source ordering depends on the target benchmark. There is no single universally best web source. The ordering changes with the benchmark. Figure 1 and the right column of Figure 3 give a concrete example. FineWeb-Edu and Nemotron-HQS are stronger on MMLU in both resume and from-scratch regimes. On PIQA, however, DCLM Baseline and Nemotron-MHQ lead among the compared sources. Nemotron-HQ is more balanced, but it still trails Nemotron-HQS on MMLU and trails Nemotron-MHQ on PIQA. The top row of Figure 4 summarizes the same pattern by averaging over shared quantiles. FineWeb-Edu and Nemotron-HQS are stronger on the knowledge-oriented group (MMLU and ARC-Easy), while DCLM and Nemotron-MHQ contribute more on the commonsense-oriented group (PIQA and HellaSwag). The practical implication is that source comparison should be benchmark-conditioned rather than treated as an absolute ranking, such as Nemotron-MHQ can be a good choice when intended for commonsense capability

Finding 2: Current scorers are strongly knowledge-biased. Prevailing scorers behave more like knowledge-oriented utility signals than benchmark-independent quality measures. On knowledge-heavy benchmarks such as MMLU and ARC-Easy, higher-score regions usually perform best or remain close to the best. *However, this trend can weaken or even reverse on broader commonsense benchmarks such as PIQA and HellaSwag.* For FineWeb-Edu, moving from the highest-score window ($q = 0$) to the $q = 60$ window decreases ARC-Easy by 8.11 points, but increases PIQA by 1.09 and HellaSwag by 1.73 (Figures 1 and 3). DCLM is similar: MMLU decreases by 2.19 and ARC-Easy decreases by 6.87, while PIQA increases by 2.24 and HellaSwag increases by 1.27. The middle row of Figure 4 makes this contrast explicit. These results show that current scorers encode a capability bias: *higher score does not always mean higher utility for every target benchmark.*

This bias has direct implications for open-web recipe design. If the target capability is not knowledge-oriented, existing scores should be used conservatively; in some score regions, they can even behave like reverse signals for the target benchmark. Recent and concurrent work has started to address scorer bias by using benchmark datasets to guide scorer training [51, 73], and prior work has also observed that educational or quality classifiers can trade retained volume against benchmark gains and over-emphasize particular text styles [59, 78]. Our contribution is to make this bias quantile-explicit: we show where along each source distribution the bias appears, how it differs across sources, and how it changes the retained fractions used by threshold-based mixing. This suggests that more balanced multi-objective scorers are important for data recipes targeting capabilities beyond knowledge-heavy evaluation, as a future direction.

Finding 3: Within-source utility variance is large. Different score quantiles within the same source can differ substantially in downstream utility. The bottom row of Figure 4 shows this directly on ARC-Easy. In the resume setting, FineWeb-Edu varies by 8.11 points between score quantiles 0 and 60. Nemotron-HQ varies by 4.76 points between score quantiles 10 and 60. The effect also appears on MMLU in Figure 3: for Nemotron-HQ, Nemotron-HQS, and DCLM, the best and worst probed quantiles differ by more than 1.5 points. At the scale of a 0.6B proxy model trained on 8.4B-token probes, this is a large downstream-utility margin. It is direct evidence that uniform-within-source sampling is a poor abstraction for modern scored corpora.

Finding 4: Ordering is reasonably stable across probing regimes. Although absolute scores differ between from-scratch and resume experiments, the relative ordering is often stable. Over shared source-quantile units, from-scratch/resume Spearman correlation reaches 0.91 for HellaSwag, 0.85 for MMLU, and 0.82 for ARC-Easy, although it is lower in others (Table 4). Both setups also agree on coarse-grained source-level trends, such as Fineweb-Edu consistently outperforming DCLM Baseline on MMLU (Figure 3). This supports using stable benchmarks as utility anchors and motivates future work to focus on the resume setting, which shows a less noisy trend.

Taken together, these findings make two points clear. First, open-web data mixing should not assume a single benchmark-independent quality standard. Second, the community's intuition about these corpora is partly right but too coarse. DCLM, FineWeb-Edu, and Nemotron do have distinct strengths, but those strengths only become visible when we examine score quantiles and target capabilities together. Full quantile plots are provided in Figures 6 and 7.

Discussion: connection to prior work. These findings help explain several observations in prior work and suggest corresponding data-centric strategies. First, stronger filtering is not necessarily better [41, 51]. When score distributions are biased toward particular capabilities, more aggressive filtering can amplify this bias and reduce average performance. Second, curriculum learning can benefit from matching data order with training dynamics [47]. Our observation of large within-source utility variance provides a data-side explanation: if high-utility regions are concentrated within a source, exposing them at the right stage can improve their effective use. Third, selective repetition can improve data efficiency when useful examples are concentrated rather than uniformly distributed [69]. This is consistent with our finding that top-ranked regions can carry disproportionate benchmark utility compared with the rest of the corpus. We test these connections in Section 7, where we adopt benchmarking-guided curriculum and selective repetition in our fully open pretraining run. We discuss the mechanistic interpretation in more detail in Section C.2.

6 From Benchmarking to Top-p Mixing

Quantile benchmarking gives discrete observations, but recipe design needs a concrete retention decision for each source. We therefore fit a source-specific utility curve for each scored source. Let $(q_{i,j}, y_{i,j})$ denote the downstream utility of source i at probed retained fractions $q_{i,j}$. Here, q is measured as a top-p percentage of the source, so a larger q means that the retained subset reaches deeper into lower-scored data. We fit a monotone decreasing surrogate $f_i(q)$ so that $y = f_i(q)$, where y is the target benchmark utility. The deduction depends on the choice of *anchor benchmark*: the downstream benchmark used as the shared utility axis for converting source-specific score quantiles into retained fractions. We discuss this choice below and test its robustness in Section F.

Table 1: Controlled 40B-token ablations. **Core** averages MMLU, ARC-C, ARC-E, and CSQA (Section D for details). **Avg.** averages the full eight-task suite reported in Table 8; this main-text table shows five representative tasks for space. The subscripts on Avg. and Core report the absolute margin over the uniform baseline. Positive margins are shown in green.

Method	Mixture	Curriculum	MMLU	ARC-C	ARC-E	CSQA	PIQA	Avg.	Core
Baseline	Uniform	Random	34.49	34.58	72.84	59.79	74.27	$55.47_{0.00}$	$50.43_{0.00}$
Top-p Mixture	Vanilla Top-p	Random	35.79	36.95	71.60	**60.28**	73.39	$55.54_{+0.07}$	$51.16_{+0.73}$
QB	Quantile-Based	Random	36.51	36.95	74.60	59.62	73.07	$55.68_{+0.21}$	$51.92_{+1.49}$
QB-Asc	Quantile-Based	Ascending	**36.65**	**38.64**	**75.66**	58.72	**74.92**	$\mathbf{56.43}_{+0.96}$	$\mathbf{52.42}_{+1.99}$

From discrete probes to an operational utility curve. The monotonicity constraint is operational rather than literal. Empirical curves can be noisy, and they can also be mildly non-monotone, especially on biased or low-signal benchmarks. We enforce monotonicity because mixture deduction needs a stable inverse map from target utility to retained fraction. The fitted curve therefore does not claim that every underlying benchmark curve is perfectly monotone. It provides an approximate surrogate for threshold selection. We give the exact constrained fitting procedure in Section F.

Global-threshold deduction. Once these curves are available, mixture deduction reduces to threshold search. Let N_i be the available token volume of source i. For a global utility threshold T, source i contributes $V_i(T) = N_i \cdot \frac{f_i^{-1}(T)}{100}$ tokens. Under a total token budget D, we iteratively search for a shared threshold T^* such that $\sum_i N_i \cdot \frac{f_i^{-1}(T^*)}{100} = D$. This yields source-specific retained fractions $q_i^* = f_i^{-1}(T^*)$ automatically. We clamp the inverse map to the feasible probed interval: out-of-range thresholds correspond to retaining either the shallowest available high-score region or the full candidate pool allowed by the recipe. If several fractions have nearly the same fitted utility, we use the shallowest feasible fraction to avoid adding lower-scored data without a clear utility gain. Figure 3 panel (b) illustrates this thresholding procedure.[4]

Choosing an anchor benchmark. The anchor is not meant to be a universal measure of data quality. It is a practical utility proxy under a constrained open-pretraining setting: the web corpora and their scorers are already given, and the remaining decision is how to expose and use their shared implicit bias. In our measurements, the shared bias is strongest for knowledge-oriented benchmarks. MMLU and ARC-Easy show more stable source ordering and more usable monotone trends than broader commonsense benchmarks such as PIQA and HellaSwag. We therefore use MMLU-oriented utility for the main threshold deduction, while reporting anchor-robustness tests in Section F. This choice predicts that gains should concentrate on knowledge-oriented metrics, a pattern we test in the controlled ablations. It is a passive choice forced by the available scorers, not the ideal end state. A better future pipeline would first specify the target capability family, then design or train scorers explicitly aligned with that utility.

7 Validation in Controlled Experiments and Open Pretraining

7.1 Controlled Validation at 40B Tokens

Step-by-step validation. We validate the framework with a Qwen3-1.7B variant and 1×Chinchilla setup, with setup details in Section D. The ablation progresses through four stages: uniform mixing as baseline, vanilla top-p mixing, quantile-benchmarking-guided mixing, and an additional ascending quality curriculum. All four runs use the same budget of approximately 41.94B token positions (nominally 40B), after downsampling or upsampling their selected pools to this common budget. Uniform mixing samples from the downsampled sources with weights proportional to source size. Vanilla top-p mixing chooses retained fractions such that the effective weights $w_i(q_i)$ in Section 3 remain proportional to source size. Quantile-benchmarking-guided mixing follows Section 6, and the curriculum version adopts the curriculum model averaging strategy detailed in Sections B and E.

This design tests three claims in order. (1) Source-internal scores are useful, measured by vanilla top-p mixing over uniform mixing. (2) Benchmark-guided threshold deduction improves over heuristic

[4]Panel (b) labels fitted within-source cutoffs, not mixture weights. In the implemented QB/QB-Asc recipe (Table 7), DCLM, FineWeb-Edu, and Nemotron have document retention rates of 13%, 31%, and 41%, but pre-resampling token mixture shares of 9.7%, 38.6%, and 51.7%, respectively. Document retention is normalized within each source; mixture share is normalized over all retained tokens across sources.

top-p filtering. (3) Curriculum can add further gain by exploiting within-source utility variance. Additional setup, Core suite evaluation, and mixture details are in Section D.

The results support this interpretation. Moving from uniform mixing to equal top-p filtering improves the Core suite by $+0.73$. Moving from equal top-p filtering to quantile-benchmarking-guided thresholds adds another $+0.76$. In this ablation, correcting the relative retained fractions gives a gain comparable to the gain from score-based filtering itself. Adding the quality curriculum yields an additional positive gain of $+0.50$.

What the trade-off tells us. The trade-off pattern is also informative. Knowledge-intensive metrics improve the most. Some broader commonsense metrics remain flat, and some slightly degrade in the full eight-task table (Table 8). This pattern is expected under the scorer-bias findings: current scorers are knowledge-biased, and the MMLU-oriented anchor benchmark optimizes knowledge-heavy utility. The validation therefore does more than report a better average. It connects the empirical findings to the behavior of the final recipe.

7.2 Curriculum and Selective Repetition

Table 2: Curriculum and selective-repetition pilots. Core averages MMLU, ARC-C, ARC-E, and CSQA. Detailed eight-task results are in Table 9.

Method	Retain	MMLU	ARC-E	PIQA	Avg.	Core
Uniform	100%	30.77	61.05	72.42	50.56	46.21
CMA	100%	31.68	**61.93**	71.71	**50.99**	**46.89**
Filter&Repeat	13.8%	**32.99**	61.75	71.71	48.87	44.14
Filter&Repeat	33.4%	32.44	**61.93**	72.09	50.83	46.65
Filter&Repeat	77.4%	31.68	60.70	**72.52**	50.49	45.83

Motivated by Section 5, we also test data-centric strategies that are connected to the utility bias-variance observations. The curriculum run follows the CMA setting [47]: it sorts data in ascending quality order and averages the last six checkpoints. The Filter&Repeat runs retain different top-p fractions and repeat them to match the baseline token budget [69]. Full setup details are in Sections D and E.

The results mirror the quantile-benchmarking findings. CMA improves the Core average, suggesting that an ascending-quality schedule can exploit within-source utility variance. Filter&Repeat improves MMLU strongly at small retained fractions, but the most aggressive setting hurts the average and Core suite. This matches the bias pattern in Section 5: concentrating on high-scored tails can amplify knowledge-oriented utility while narrowing the data distribution. These supportive pilots motivate carrying curriculum and the moderate Filter&Repeat strategy into the final open pretraining recipe, since they provide benefits on the intended capabilities and show moderately better average performance.

7.3 Open 2B Pretraining

Data recipe in fully open pretraining. We next apply the resulting recipe to the fully open KAIYUAN-2B pretraining run. The model serves as a practical transfer test of the framework and as a fully open release case.

The final recipe uses several practical extensions, including approximate top-p mixing, selective repetition, and late-stage curriculum with model averaging, discussed in prior sections. We also adopt phase-wise domain mixture evolution following prior work [3]. These ingredients are not individually new. Instead, the 2B run applies the data-valuation view suggested by quantile benchmarking, and supported by the controlled ablations, into a practical scaling setting. We adapt heterogeneous mixing on FineWeb-Edu and DCLM Baseline, resulting in the phase mixture shown in Table 13. We provide the concrete curriculum and selective-repetition construction in Section E.

On targeted Chinese, mathematics, and coding benchmarks, KAIYUAN-2B achieves an average of 46.05. In Figure 2, it reaches the parameter-efficiency Pareto frontier among similarly scaled fully open baselines in our comparison, and it remains competitive with larger models in focused domains. On reasoning and knowledge benchmarks, KAIYUAN-2B averages 67.74. This matches YuLan-Mini-2.4B while using fewer effective backbone parameters, and it outperforms SmolLM2-1.7B within the fully open group. The compact comparison in Figure 2 previews this release result, while full benchmark tables, parameter breakdowns, and evaluation settings are provided in Sections H and J. The datasets, training recipe, and data preprocessing framework are all open-source.

8 Conclusion

We first model the open-web pretraining data design into a heterogeneous top-p mixing problem. We then propose quantile benchmarking to make this decision measurable by mapping source-specific score regions onto a shared benchmark utility scale. Our measurements reveal benchmark-dependent source ordering, large within-source utility variance, and a clear knowledge bias in current scorers; threshold deduction turns these observations into source-specific retained fractions. Controlled ablations validate the recipe, and the fully open KAIYUAN-2B run demonstrates an end-to-end application. We release the model, recipe, and diagnostics so that these data decisions are inspectable and reproducible.

Acknowledgments and Disclosure of Funding

This work was partially supported by the Major Key Project of PCL under Grant PCL2025A12.

References

[1] Marah I Abdin, Jyoti Aneja, Harkirat S. Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report. *CoRR*, abs/2412.08905, 2024. doi: 10.48550/ARXIV.2412.08905. URL `https://doi.org/10.48550/arXiv.2412.08905`.

[2] Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp A. Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone. *CoRR*, abs/2404.14219, 2024. doi: 10.48550/ARXIV.2404.14219. URL `https://doi.org/10.48550/arXiv.2404.14219`.

[3] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlícek, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big - data-centric training of a small language model. *CoRR*, abs/2502.02737, 2025. doi: 10.48550/ARXIV.2502.02737. URL `https://doi.org/10.48550/arXiv.2502.02737`.

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

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

[6] 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. Program synthesis with large language models. *CoRR*, abs/2108.07732, 2021. URL `https://arxiv.org/abs/2108.07732`.

[7] Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics, 2023.

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

[9] 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, Anna Shors, Anubhav Mandarwal, Arham Mehta, Arun Venkatesan, Ashton Sharabiani, Ashwath Aithal, Ashwin Poojary, Ayush Dattagupta, Balaram Buddharaju, Banghua Zhu, Barnaby Simkin, Bilal Kartal, Bita Darvish Rouhani, Bobby Chen, Boris Ginsburg, Brandon Norick, Brian Yu, Bryan Catanzaro, Charles Wang, Charlie Truong, Chetan Mungekar, Chintan Patel, Chris Alexiuk, Christian Munley, Christopher Parisien, Dan Su, Daniel Afrimi, Daniel Korzekwa, Daniel Rohrer, Daria Gitman, David Mosallanezhad, Deepak Narayanan, Dima Rekesh, Dina Yared, Dmytro Pykhtar, Dong Ahn, Duncan Riach, Eileen Long, Elliott Ning, Eric Chung, Erick Galinkin, Evelina Bakhturina, Gargi Prasad, Gerald Shen, Haifeng Qian, Haim Elisha, Harsh Sharma, Hayley Ross, Helen Ngo, Herman Sahota, Hexin Wang, Hoo Chang Shin, Hua Huang, Iain Cunningham, Igor Gitman, Ivan Moshkov, Jaehun Jung, Jan Kautz, Jane Polak Scowcroft, Jared Casper, Jian Zhang, Jiaqi Zeng, Jimmy Zhang, Jinze Xue, Jocelyn Huang, Joey Conway, John Kamalu, Jonathan M. Cohen, Joseph Jennings, Julien Veron Vialard, Junkeun Yi, Jupinder Parmar, Kari Briski, Katherine Cheung, Katherine Luna, Keith W. Ross, Keshav Santhanam, Kezhi Kong, Krzysztof Pawelec, and Kumar Anik. NVIDIA nemotron nano 2: An accurate and efficient hybrid mamba-transformer reasoning model. *CoRR*, abs/2508.14444, 2025. doi: 10.48550/ARXIV.2508.14444. URL `https://doi.org/10.48550/arXiv.2508.14444`.

[10] Lior Belenki, Alekh Agarwal, Tianze Shi, and Kristina Toutanova. Optimizing pre-training data mixtures with mixtures of data expert models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32570–32587, 2025.

[11] Marco Bellagente, Jonathan Tow, Dakota Mahan, Duy Phung, Maksym Zhuravinskyi, Reshinth Adithyan, James Baicoianu, Ben Brooks, Nathan Cooper, Ashish Datta, Meng Lee, Emad Mostaque, Michael Pieler, Nikhil Pinnaparju, Paulo Rocha, Harry Saini, Hannah Teufel, Niccolo Zanichelli, and Carlos Riquelme. Stable lm 2 1.6b technical report. *arXiv preprint arXiv:2402.17834*, 2024. URL `https://arxiv.org/abs/2402.17834`.

[12] Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. Smollm-corpus, July 2024. URL `https://huggingface.co/datasets/HuggingFaceTB/smollm-corpus`.

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

[14] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 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, 2020. doi: 10.1609/AAAI.V34I05.6239. URL `https://doi.org/10.1609/aaai.v34i05.6239`.

[15] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul

Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021. URL https://arxiv.org/abs/2107.03374.

[16] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

[17] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018. URL http://arxiv.org/abs/1803.05457.

[18] 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. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

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

[20] OpenCSG Community. 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, 2024. [Accessed 03-12-2025].

[21] Together Computer. Redpajama: An open source recipe to reproduce llama training dataset, April 2023. URL https://github.com/togethercomputer/RedPajama-Data.

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

[23] Kazuki Fujii, Yukito Tajima, Sakae Mizuki, Hinari Shimada, Taihei Shiotani, Koshiro Saito, Masanari Ohi, Masaki Kawamura, Taishi Nakamura, Takumi Okamoto, Shigeki Ishida, Kakeru Hattori, Youmi Ma, Hiroya Takamura, Rio Yokota, and Naoaki Okazaki. Rewriting pre-training data boosts llm performance in math and code, 2025. URL https://arxiv.org/abs/2505.02881.

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

[25] Yuling Gu, Oyvind Tafjord, Bailey Kuehl, Dany Haddad, Jesse Dodge, and Hannaneh Hajishirzi. OLMES: A standard for language model evaluations. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *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, 2025. doi: 10.18653/V1/2025.FINDINGS-NAACL.282. URL https://doi.org/10.18653/v1/2025.findings-naacl.282.

[26] Jie Hao, Rui Yu, Wei Zhang, Huixia Wang, Jie Xu, and Mingrui Liu. BLISS: A lightweight bilevel influence scoring method for data selection in language model pretraining. *arXiv preprint arXiv:2510.06048*, 2025.

[27] Conghui He, Zhenjiang Jin, Chao Xu, Jiantao Qiu, Bin Wang, Wei Li, Hang Yan, Jiaqi Wang, and Dahua Lin. Wanjuan: A comprehensive multimodal dataset for advancing english and chinese large models, 2023.

[28] Conghui He, Wei Li, Zhenjiang Jin, Chao Xu, Bin Wang, and Dahua Lin. Opendatalab: Empowering general artificial intelligence with open datasets, 2024. URL https://arxiv.org/abs/2407.13773.

[29] David Heineman, Valentin Hofmann, Ian Magnusson, Yuling Gu, Noah A. Smith, Hannaneh Hajishirzi, Kyle Lo, and Jesse Dodge. Signal and noise: A framework for reducing uncertainty in language model evaluation. *CoRR*, abs/2508.13144, 2025. doi: 10.48550/ARXIV.2508.13144. URL https://doi.org/10.48550/arXiv.2508.13144.

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

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

[32] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Joaquin Vanschoren and Sai-Kit Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

[33] Shengding Hu, Yuge Tu, Xu Han, Ganqu Cui, Chaoqun He, Weilin Zhao, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Xinrong Zhang, Zhen Leng Thai, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, dahai li, Zhiyuan Liu, and Maosong Sun. MiniCPM: Unveiling the potential of small language models with scalable training strategies. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=3X2L2TFr0f.

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

[35] 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. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *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*, 2023. URL http://papers.nips.cc/paper%5Ffiles/paper/2023/hash/c6ec1844bec96d6d32ae95ae694e23d8-Abstract-Datasets%5Fand%5FBenchmarks.html.

[36] Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 1(3):3, 2023.

[37] 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. The stack: 3 tb of permissively licensed source code. *Preprint*, 2022.

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

[39] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. `https://allenai.org/tulu`, 2024.

[40] Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. CMMLU: measuring massive multitask language understanding in chinese. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *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, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.671. URL `https://doi.org/10.18653/v1/2024.findings-acl.671`.

[41] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Kumar Guha, Sedrick Scott Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee F. Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah M. Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Raghavi Chandu, Thao Nguyen, Igor Vasiljevic, Sham M. Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alex Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. DataComp-LM: In search of the next generation of training sets for language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL `http://papers.nips.cc/paper%5Ffiles/paper/2024/hash/19e4ea30dded58259665db375885e412-Abstract-Datasets%5Fand%5FBenchmarks%5FTrack.html`.

[42] Shengrui Li, Fei Zhao, Kaiyan Zhao, Jieying Ye, Haifeng Liu, Fangcheng Shi, Zheyong Xie, Yao Hu, and Shaosheng Cao. Decouple searching from training: Scaling data mixing via model merging for large language model pre-training. *arXiv preprint arXiv:2602.00747*, 2026.

[43] Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. Regmix: Data mixture as regression for language model pre-training. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=5BjQOUXq7i`.

[44] 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. The flan collection: Designing data and methods for effective instruction tuning, 2023.

[45] Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, Tianyang Liu, Max Tian, Denis Kocetkov, Arthur Zucker, Younes Belkada, Zijian Wang, Qian Liu, Dmitry Abulkhanov,

Indraneil Paul, Zhuang Li, Wen-Ding Li, Megan Risdal, Jia Li, Jian Zhu, Terry Yue Zhuo, Evgenii Zheltonozhskii, Nii Osae Osae Dade, Wenhao Yu, Lucas Krauß, Naman Jain, Yixuan Su, Xuanli He, Manan Dey, Edoardo Abati, Yekun Chai, Niklas Muennighoff, Xiangru Tang, Muhtasham Oblokulov, Christopher Akiki, Marc Marone, Chenghao Mou, Mayank Mishra, Alex Gu, Binyuan Hui, Tri Dao, Armel Zebaze, Olivier Dehaene, Nicolas Patry, Canwen Xu, Julian McAuley, Han Hu, Torsten Scholak, Sebastien Paquet, Jennifer Robinson, Carolyn Jane Anderson, Nicolas Chapados, Mostofa Patwary, Nima Tajbakhsh, Yacine Jernite, Carlos Mu noz Ferrandis, Lingming Zhang, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. Starcoder 2 and the stack v2: The next generation, 2024.

[46] Kairong Luo, Haodong Wen, Shengding Hu, Zhenbo Sun, Zhiyuan Liu, Maosong Sun, Kaifeng Lyu, and Wenguang Chen. A Multi-Power law for loss curve prediction across learning rate schedules. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=KnoS9XxIlK.

[47] Kairong Luo, Zhenbo Sun, Haodong Wen, Xinyu Shi, Jiarui Cui, Chenyi Dang, Kaifeng Lyu, and Wenguang Chen. How learning rate decay wastes your best data in curriculum-based LLM pretraining. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=T5wkZJqzkz.

[48] Sachin Mehta, Mohammad Hossein Sekhavat, Qingqing Cao, Maxwell Horton, Yanzi Jin, Chenfan Sun, Iman Mirzadeh, Mahyar Najibi, Dmitry Belenko, Peter Zatloukal, et al. OpenELM: An efficient language model family with open training and inference framework. *arXiv preprint arXiv:2404.14619*, 2024.

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

[50] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium, oct 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

[51] David Mizrahi, Anders Boesen Lindbo Larsen, Jesse Allardice, Suzie Petryk, Yuri Gorokhov, Jeffrey Li, Alex Fang, Josh Gardner, Tom Gunter, and Afshin Dehghan. Language models improve when pretraining data matches target tasks. *CoRR*, abs/2507.12466, 2025. doi: 10.48550/ARXIV.2507.12466. URL https://doi.org/10.48550/arXiv.2507.12466.

[52] Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A. Raffel. Scaling data-constrained language models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *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*, 2023.

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

[54] 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, Ameya Sunil Mahabaleshwarkar, Amir Klein, Amit Zuker, Amnon Geifman, Amy Shen, Anahita Bhiwandiwalla, Andrew Tao, Anjulie Agrusa, Ankur Verma, Ann Guan, Anubhav Mandarwal, Arham Mehta, Ashwath Aithal, Ashwin Poojary, Asif Ahamed, Asit Mishra, Asma Kuriparambil Thekkumpate, Ayush Dattagupta, Banghua Zhu, Bardiya Sadeghi, Barnaby Simkin, Ben Lanir, Benedikt Schifferer, Besmira Nushi, Bilal Kartal, Bita Darvish Rouhani, Boris Ginsburg, Brandon Norick, Brandon Soubasis, Branislav Kisacanin, Brian Yu, Bryan Catanzaro, Carlo del Mundo, Chantal Hwang, Charles Wang,

Cheng-Ping Hsieh, Chenghao Zhang, Chenhan Yu, Chetan Mungekar, Chintan Patel, Chris Alexiuk, Christopher Parisien, Collin Neale, Cyril Meurillon, Damon Mosk-Aoyama, Dan Su, Dane Corneil, Daniel Afrimi, Daniel Lo, Daniel Rohrer, Daniel Serebrenik, Daria Gitman, Daria Levy, Darko Stosic, David Mosallanezhad, Deepak Narayanan, Dhruv Nathawani, Dima Rekesh, Dina Yared, Divyanshu Kakwani, Dong Ahn, Duncan Riach, Dusan Stosic, Edgar Minasyan, Edward Lin, Eileen Long, Eileen Peters Long, Elad Segal, Elena Lantz, Ellie Evans, Elliott Ning, Eric Chung, Eric Harper, Eric Tramel, Erick Galinkin, Erik Pounds, Evan Briones, Evelina Bakhturina, Evgeny Tsykunov, Faisal Ladhak, Fay Wang, Fei Jia, Felipe Soares, Feng Chen, Ferenc Galko, Frank Sun, Frankie Siino, Gal Hubara Agam, Ganesh Ajjanagadde, Gantavya Bhatt, Gargi Prasad, George Armstrong, Gerald Shen, Gorkem Batmaz, Grigor Nalbandyan, Haifeng Qian, Harsh Sharma, Hayley Ross, Helen Ngo, Herbert Hum, Herman Sahota, Hexin Wang, Himanshu Soni, Hiren Upadhyay, Huizi Mao, Huy C Nguyen, Huy Q Nguyen, Iain Cunningham, Ido Galil, Ido Shahaf, Igor Gitman, Ilya Loshchilov, Itamar Schen, Itay Levy, Ivan Moshkov, Izik Golan, Izzy Putterman, Jan Kautz, Jane Polak Scowcroft, Jared Casper, Jatin Mitra, Jeffrey Glick, Jenny Chen, Jesse Oliver, Jian Zhang, Jiaqi Zeng, Jie Lou, Jimmy Zhang, Jinhang Choi, Jining Huang, Joey Conway, Joey Guman, John Kamalu, Johnny Greco, Jonathan Cohen, Joseph Jennings, Joyjit Daw, Julien Veron Vialard, Junkeun Yi, Jupinder Parmar, Kai Xu, Kan Zhu, Kari Briski, Katherine Cheung, Katherine Luna, Keith Wyss, Keshav Santhanam, Kevin Shih, Kezhi Kong, Khushi Bhardwaj, Kirthi Shankar, Krishna C. Puvvada, Krzysztof Pawelec, Kumar Anik, Lawrence McAfee, Laya Sleiman, Leon Derczynski, Li Ding, Lizzie Wei, Lucas Liebenwein, Luis Vega, Maanu Grover, Maarten Van Segbroeck, Maer Rodrigues de Melo, Mahdi Nazemi, Makesh Narsimhan Sreedhar, Manoj Kilaru, Maor Ashkenazi, Marc Romeijn, Marcin Chochowski, Mark Cai, Markus Kliegl, Maryam Moosaei, Matt Kulka, Matvei Novikov, Mehrzad Samadi, Melissa Corpuz, Mengru Wang, Meredith Price, Michael Andersch, Michael Boone, Michael Evans, Miguel Martinez, Mikail Khona, Mike Chrzanowski, Minseok Lee, Mohammad Dabbah, Mohammad Shoeybi, Mostofa Patwary, Nabin Mulepati, Najeeb Nabwani, Natalie Hereth, Nave Assaf, Negar Habibi, Neta Zmora, Netanel Haber, Nicola Sessions, Nidhi Bhatia, Nikhil Jukar, Nikki Pope, Nikolai Ludwig, Nima Tajbakhsh, Nir Ailon, Nirmal Juluru, Nishant Sharma, Oleksii Hrinchuk, Oleksii Kuchaiev, Olivier Delalleau, Oluwatobi Olabiyi, Omer Ullman Argov, Omri Puny, Oren Tropp, Ouye Xie, Parth Chadha, Pasha Shamis, Paul Gibbons, Pavlo Molchanov, Pawel Morkisz, Peter Dykas, Peter Jin, Pinky Xu, Piotr Januszewski, Pranav Prashant Thombre, Prasoon Varshney, Pritam Gundecha, Przemek Tredak, Qing Miao, Qiyu Wan, Rabeeh Karimi Mahabadi, Rachit Garg, Ran El-Yaniv, Ran Zilberstein, Rasoul Shafipour, Rich Harang, Rick Izzo, Rima Shahbazyan, Rishabh Garg, Ritika Borkar, Ritu Gala, Riyad Islam, Robert Hesse, Roger Waleffe, Rohit Watve, Roi Koren, Ruoxi Zhang, Russell Hewett, Russell J. Hewett, Ryan Prenger, Ryan Timbrook, Sadegh Mahdavi, Sahil Modi, Samuel Kriman, Sangkug Lim, Sanjay Kariyappa, Sanjeev Satheesh, Saori Kaji, Satish Pasumarthi, Saurav Muralidharan, Sean Narentharen, Sean Narenthiran, Seonmyeong Bak, Sergey Kashirsky, Seth Poulos, Shahar Mor, Shanmugam Ramasamy, Shantanu Acharya, Shaona Ghosh, Sharath Turuvekere Sreenivas, Shelby Thomas, Shiqing Fan, Shreya Gopal, Shrimai Prabhumoye, Shubham Pachori, Shubham Toshniwal, Shuoyang Ding, Siddharth Singh, Simeng Sun, Smita Ithape, Somshubra Majumdar, Soumye Singhal, Stas Sergienko, Stefania Alborghetti, Stephen Ge, Sugam Dipak Devare, Sumeet Kumar Barua, Suseella Panguluri, Suyog Gupta, Sweta Priyadarshi, Syeda Nahida Akter, Tan Bui, Teodor-Dumitru Ene, Terry Kong, Thanh Do, Tijmen Blankevoort, Tim Moon, Tom Balough, Tomer Asida, Tomer Bar Natan, Tomer Ronen, Tugrul Konuk, Twinkle Vashishth, Udi Karpas, Ushnish De, Vahid Noorozi, Vahid Noroozi, Venkat Srinivasan, Venmugil Elango, Victor Cui, Vijay Korthikanti, Vinay Rao, Vitaly Kurin, Vitaly Lavrukhin, Vladimir Anisimov, Wanli Jiang, Wasi Uddin Ahmad, Wei Du, Wei Ping, Wenfei Zhou, Will Jennings, William Zhang, Wojciech Prazuch, Xiaowei Ren, Yashaswi Karnati, Yejin Choi, Yev Meyer, Yi-Fu Wu, Yian Zhang, Yigong Qin, Ying Lin, Yonatan Geifman, Yonggan Fu, Yoshi Subara, Yoshi Suhara, Yubo Gao, Zach Moshe, Zhen Dong, Zhongbo Zhu, Zihan Liu, Zijia Chen, and Zijie Yan. Nvidia nemotron 3: Efficient and open intelligence. *CoRR*, 2025. URL `https://arxiv.org/abs/2512.20856`.

[55] Beijing Academy of Artificial Intelligence. Chinese corpus internet usage agreement. `https://data.baai.ac.cn/resources/agreement/cci_usage_aggrement.pdf`, 2023. [Accessed 03-12-2025].

[56] Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. Openwebmath: An open dataset of high-quality mathematical web text, 2023.

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

[58] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data only. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *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*, 2023. URL http://papers.nips.cc/paper%5Ffiles/paper/2023/hash/fa3ed726cc5073b9c31e3e49a807789c-Abstract-Datasets%5Fand%5FBenchmarks.html.

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

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

[61] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 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, 2020. doi: 10.1609/AAAI.V34I05.6399. URL https://doi.org/10.1609/aaai.v34i05.6399.

[62] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454/.

[63] Xiaofeng Shi, Lulu Zhao, Hua Zhou, and Donglin Hao. Industrycorpus2, 2024. URL https://huggingface.co/datasets/BAAI/IndustryCorpus2.

[64] KaShun SHUM, Yuzhen Huang, Hongjian Zou, dingqi, YiXuan Liao, Xiaoxin Chen, Qian Liu, and Junxian He. Predictive data selection: The data that predicts is the data that teaches. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=tTVYR82Iz6.

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

[66] Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-CC: Transforming Common Crawl into a refined long-horizon pretraining dataset. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *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, 2025. URL https://aclanthology.org/2025.acl-long.123/.

[67] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C), February 2024. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

[68] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

[69] Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari Morcos. D4: improving LLM pretraining via document de-duplication and diversification. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *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*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/a8f8cbd7f7a5fb2c837e578c75e5b615-Abstract-Datasets_and_Benchmarks.html.

[70] 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, Michal Guerquin, David Heineman, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James Validad Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Jake Poznanski, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 OLMo 2 furious (COLM's version). In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=2ezugTT9kU.

[71] Jiapeng Wang, Changxin Tian, Kunlong Chen, Ziqi Liu, Jiaxin Mao, Wayne Xin Zhao, Zhiqiang Zhang, and Jun Zhou. MergeMix: Optimizing mid-training data mixtures via learnable model merging. *arXiv preprint arXiv:2601.17858*, 2026.

[72] Liangdong Wang, Bo-Wen Zhang, Chengwei Wu, Hanyu Zhao, Xiaofeng Shi, Shuhao Gu, Jijie Li, Quanyue Ma, TengFei Pan, and Guang Liu. Cci3.0-hq: a large-scale chinese dataset of high quality designed for pre-training large language models, 2024. URL https://arxiv.org/abs/2410.18505.

[73] Shaobo Wang, Xuan Ouyang, Tianyi Xu, Yuzheng Hu, Jialin Liu, Guo Chen, Tianyu Zhang, Junhao Zheng, Kexin Yang, Xingzhang Ren, et al. Opus: Towards efficient and principled data selection in large language model pre-training in every iteration. *arXiv preprint arXiv:2602.05400*, 2026.

[74] Yifan Wang, Binbin Liu, Fengze Liu, Yuanfan Guo, Jiyao Deng, Xuecheng Wu, Weidong Zhou, Xiaohuan Zhou, and Taifeng Wang. TiKMiX: Take data influence into dynamic mixture for language model pre-training. *arXiv preprint arXiv:2508.17677*, 2025.

[75] Yudong Wang, Zixuan Fu, Jie Cai, Peijun Tang, Hongya Lyu, Yewei Fang, Zhi Zheng, Jie Zhou, Guoyang Zeng, Chaojun Xiao, et al. Ultra-fineweb: Efficient data filtering and verification for high-quality llm training data. *arXiv preprint arXiv:2505.05427*, 2025.

[76] Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, Haihua Yang, Biye Li, Cheng Cheng, Weiwei Lü, Rui Hu, Chenxia Li, Liu Yang, Xilin Luo, Xuejie Wu, Lunan Liu, Wenjun Cheng, Peng Cheng, Jianhao Zhang, Xiaoyu Zhang, Lei Lin, Xiaokun Wang, Yutuan Ma, Chuanhai Dong, Yanqi Sun, Yifu Chen, Yongyi Peng, Xiaojuan Liang, Shuicheng Yan, Han Fang, and Yahui Zhou. Skywork: A more open bilingual foundation model, 2023.

[77] Kaiyue Wen, David Leo Wright Hall, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=2J51qUZ0iG.

[78] Alexander Wettig, Kyle Lo, Sewon Min, Hannaneh Hajishirzi, Danqi Chen, and Luca Soldaini. Organize the web: Constructing domains enhances pre-training data curation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=boSqwdvJVC.

[79] Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 69798–69818. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/dcba6be91359358c2355cd920da3fcbd-Paper-Conference.pdf.

[80] Tingkai Yan, Haodong Wen, Binghui Li, Kairong Luo, Wenguang Chen, and Kaifeng Lyu. Larger datasets can be repeated more: A theoretical analysis of multi-epoch scaling in linear regression, 2025.

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

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

[83] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

[84] Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=jjCB27TMK3.

[85] 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. YuLan-mini: Pushing the limits of open data-efficient language model. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5374–5400, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.268. URL https://aclanthology.org/2025.acl-long.268/.

[86] Yijiong Yu, Ziyun Dai, Zekun Wang, Wei Wang, Ran Chen, and Ji Pei. Opencsg chinese corpus: A series of high-quality chinese datasets for llm training, 2025. URL https://arxiv.org/abs/2501.08197.

[87] Zichun Yu, Fei Peng, Jie Lei, Arnold Overwijk, Wen tau Yih, and Chenyan Xiong. Group-level data selection for efficient pretraining. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=uX4dyc7Z5Z.

[88] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

[89] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.

[90] Yifan Zhang, Yifan Luo, Yang Yuan, and Andrew C Yao. Autonomous data selection with zero-shot generative classifiers for mathematical texts. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4168–4189, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.216. URL https://aclanthology.org/2025.findings-acl.216/.

[91] 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. Cpm-2: Large-scale cost-effective pre-trained language models. *AI Open*, 2:216–224, 2021. ISSN 2666-6510. doi: https://doi.org/10.1016/j.aiopen.2021.12.003. URL https://www.sciencedirect.com/science/article/pii/S2666651021000310.

[92] Zhengyan Zhang, Xu Han, Hao Zhou, Pei Ke, Yuxian Gu, Deming Ye, Yujia Qin, Yusheng Su, Haozhe Ji, Jian Guan, Fanchao Qi, Xiaozhi Wang, Yanan Zheng, Guoyang Zeng, Huanqi Cao, Shengqi Chen, Daixuan Li, Zhenbo Sun, Zhiyuan Liu, Minlie Huang, Wentao Han, Jie Tang, Juanzi Li, Xiaoyan Zhu, and Maosong Sun. Cpm: A large-scale generative chinese pre-trained language model. *AI Open*, 2:93–99, 2021. ISSN 2666-6510. doi: https://doi.org/10.1016/

j.aiopen.2021.07.001. URL https://www.sciencedirect.com/science/article/pii/S266665102100019X.

[93] Fan Zhou, Zengzhi Wang, Nikhil Ranjan, Zhoujun Cheng, Liping Tang, Guowei He, Zhengzhong Liu, and Eric P. Xing. Megamath: Pushing the limits of open math corpora. *CoRR*, abs/2504.02807, 2025. doi: 10.48550/ARXIV.2504.02807. URL https://doi.org/10.48550/arXiv.2504.02807.

[94] Kun Zhou, Beichen Zhang, Jiapeng Wang, Zhipeng Chen, Wayne Xin Zhao, Jing Sha, Zhichao Sheng, Shijin Wang, and Ji-Rong Wen. Jiuzhang3.0: Efficiently improving mathematical reasoning by training small data synthesis models. https://huggingface.co/datasets/ToheartZhang/JiuZhang3.0-Corpus-CoT, 2024.

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

Appendix Contents

A Limitations and Discussion **23**

B Additional Related Work **23**

C Quantile Benchmarking Results and Discussions **25**

C.1 Detailed Quantile Benchmarking Results 25

C.2 Discussion: Connection to Prior Work 26

D Detailed Ablation Results **29**

E Curriculum and Selective Repetition Details **30**

E.1 Multi-Dataset Curriculum Construction 30

E.2 Phase-Wise Selective Repetition 31

E.3 Instruction-Style Data in Late Pretraining 31

F Utility Fitting and Mixture Deduction **33**

G Recipe Details and Phase-wise Mixtures **34**

H Training and Evaluation Details **36**

H.1 Training Configuration 36

H.2 Evaluation Setup 39

I Release and Dataset Details **41**

J Full Benchmark and Parameter Tables **43**

A Limitations and Discussion

The main limitation of our framework is benchmark dependence. Quantile-benchmarking-guided thresholding optimizes the recipe for a target capability proxy. It does not optimize for an all-purpose notion of quality. This is why knowledge-oriented anchors produce stronger gains on MMLU and ARC than on broader commonsense tasks.

A second limitation is that current web scorers are mostly one-dimensional. They compress many aspects of data quality into one local score. This makes scorer bias hard to avoid. Our findings suggest that future recipe design should move toward multi-objective or multi-dimensional scoring rather than searching for a single benchmark-independent quality number.

A third limitation is validation scale. Our strongest disentangled evidence comes from controlled 40B-token experiments. The fully open 2B run provides an end-to-end transfer case. However, we do not exhaustively rerun trillion-token training under multiple seeds or many alternative anchor benchmarks because of compute cost.

A fourth limitation is release compatibility. Some sources, such as Nemotron, perform well under proxy benchmarking but are not compatible with a fully redistributable open-data release. This means that the best benchmark-guided source is not always the one that can appear in a fully open final recipe.

Broader impacts. Our framework can help academic groups study pretraining data recipes with lower cost and better reproducibility. This is a positive outcome for open research. At the same time, any method that improves general-purpose language model pretraining can also strengthen systems that may be misused for harmful text generation. In addition, benchmark-conditioned recipe design can amplify existing scorer bias if it is applied without care. These risks further motivate better scorer design, clearer licensing practice, and more balanced multi-objective data selection.

Future directions. These limitations also point to future work. A natural next step is multi-objective threshold deduction that explicitly balances knowledge-heavy, commonsense, multilingual, code, and math targets. Another is to measure ordering stability more quantitatively, for example through rank correlation across probing regimes and benchmark families. A third is to study how quantile benchmarking should interact with multi-phase scheduling decisions when the final recipe includes curriculum, repetition, and domain transitions simultaneously.

AI assistant usage. AI tools were used only for writing support. They were not part of the scientific method or the experimental pipeline.

B Additional Related Work

Fully open pretraining and curated web corpora. Fully open pretraining efforts have made language-model recipes increasingly reproducible by releasing model weights, training data, code, logs, ablation protocols, and detailed data compositions [3, 8, 9, 24, 34, 54, 70, 85, 95]. These reports show that modern open recipes are built from mixtures of curated web text and more targeted sources such as code, mathematics, scientific text, multilingual text, and instruction-style data. They also reveal a persistent gap between fully open models and strong open-weight systems such as Qwen [81–83], making recipe design and data use increasingly central.

Additionally, earlier efforts such as Pythia [13], TinyLlama [89], StableLM 2 [11], OpenELM [48], MiniCPM [33], and Phi [1, 2, 36] also helped establish the value of transparent small- and medium-scale pretraining. These works differ from ours mainly in focus. Many emphasize model release or reproducibility of a full pipeline, while we focus on the specific scientific problem of how to compare and mix scored open-web corpora.

Open curated web corpora are a major part of this design space. RedPajama, RefinedWeb, DCLM Baseline, FineWeb-Edu, Nemotron-CC, and related datasets provide large processed web collections, often together with filtering pipelines, quality annotations, or classifier scores [21, 41, 58, 59, 64, 66]. They have become common infrastructure for fully open pretraining, data-mixing studies, and recipe ablations [47, 70, 77, 78]. However, existing reports mostly document the final mixtures that were

used, rather than provide a general way to decide how much of each scored web corpus should be retained. Our work focuses on this missing design problem: given several large curated web sources with their own scores, how should a recipe choose source-specific top-p retained fractions under a fixed token budget?

Data curation, scoring, and filtering. Data curation consistently improves open-web pretraining quality. Prior work studies heuristic filtering, learned quality classifiers, proxy-model selection, classifier ensembling, and selective refinement [9, 41, 51, 59, 64, 66]. These works are important because they show that scorer-guided filtering can produce better training corpora. In most cases, however, the scorer is evaluated by selecting a high-score subset, training on the filtered data, and comparing the resulting model with models trained on less-filtered or differently filtered data. This establishes that filtering helps, but it gives a relatively coarse view of what the scorer prefers.

Several observations in prior work already suggest that scorer behavior is not one-dimensional. For example, different filtering pipelines can preserve different regions of the web distribution, and classifier-based curation can trade off benchmark quality against retained token volume [41, 59, 66]. Such results implicitly point to scorer bias and source mismatch. Our work makes these effects explicit and systematic. Instead of evaluating only one retained top-p subset, quantile benchmarking probes multiple score quantiles from each source and evaluates the resulting proxy checkpoints. This lets us measure benchmark-dependent source ordering, within-source utility variance, and scorer bias across the score distribution. In this sense, our goal is not to propose another web filter, but to evaluate and calibrate existing corpus-provided scores so that they can be used more reliably in open recipe design.

Data mixing and selection methods. Our work is also related to data mixture optimization. Methods such as DoReMi, RegMix, and data mixing laws estimate how different source mixtures affect downstream or language-modeling performance and use this signal to optimize domain or dataset weights [43, 79, 84]. Recent efficient mixture-search methods, including MDE, DeMix/MergeMix, TiKMiX, and related approaches, further reduce the cost of mixture search by using proxy models, regression, online adaptation, or model merging to infer better data ratios [10, 42, 71, 74]. These methods are highly relevant, but they usually take the candidate domains or datasets as the units of optimization. In other words, they ask how much weight to assign to each source, often assuming that sampling within each source is uniform or already fixed.

Our setting changes the interface of the problem. For scored web corpora, the key decision is not only the source-level mixture weight, but also which top-scoring fraction of each source is retained before mixing. This matters because curated web corpora are not internally homogeneous: different score regions within the same source can support different downstream capabilities. Heterogeneous top-p mixing therefore treats filtering and mixing as a coupled problem. The source-specific retained fraction determines both the internal quality profile of the retained subset and its contribution to the final token budget.

Influence-based and group-level data selection methods, such as Group-MATES, BLISS, and related learned selection approaches, are also connected to our work [26, 87]. These methods learn model-aware scores or group utilities for selecting training data. In addition, the proxy experiment framework is related to prior work. Wang et al. [75] also proposes to evaluate data by resuming from a checkpoint. By contrast, our method starts from a practical constraint common in open pretraining: large web corpora already come with source-specific scores, and these scores are not calibrated to one another. Quantile benchmarking is therefore complementary to mixture search and learned selection. It first turns heterogeneous within-source scores into benchmark-conditioned utility estimates; standard mixture-search or selection methods could then be applied on top of the retained subsets. For this reason, we compare primarily against uniform mixing and heuristic top-p filtering, which are the direct baselines for the filtering-and-mixing interface studied here, while discussing broader mixture-search methods as complementary tools rather than drop-in replacements.

Curriculum and repetition. Our work also connects to curriculum learning, repeated exposure, and late-stage data scheduling [47, 52, 69, 80]. Luo et al. [47] identify a mismatch between curriculum schedules and training dynamics, and propose to use model averaging to replace learning rate decay or use moderate LR decay. Tirumala et al. [69] investigates improving data efficiency by deduplication and selectively repeating high-quality data. We do not claim these ingredients are individually

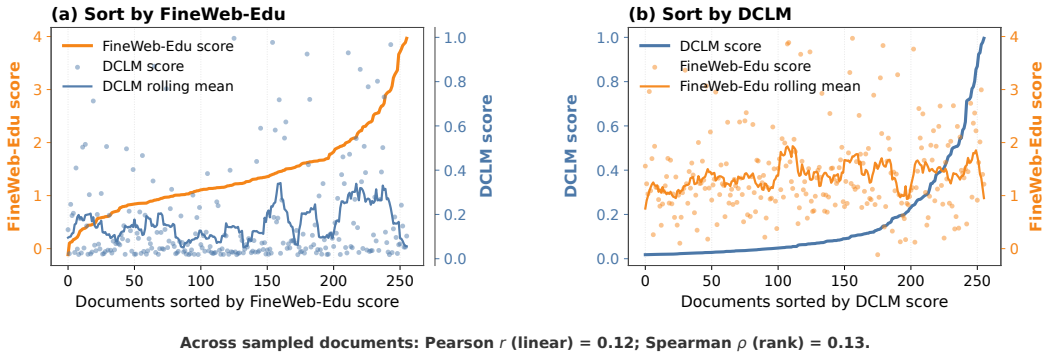

Figure 5: **FineWeb-Edu and DCLM scores are almost uncorrelated on a shared document sample.** In each panel, the x-axis orders the same documents by one scorer. The left y-axis shows the sorting scorer, which becomes monotone by construction. The right y-axis shows the other scorer at the same document indices, which remains close to noise rather than following the sorted trend. This illustrates why raw scores from different web corpora should not be compared with a shared threshold. The rolling mean uses convolution with a window length of 13. Similar observations are reported in Su et al. [66].

Table 3: Training hyperparameter configurations for quantile benchmarking.

Training from Scratch

Parameter	Value
Learning Rate	1×10^{-3}
Batch Size	512 sequences
Warmup Steps	400
Total Steps	4,000

Continual Training

Parameter	Value
Peak Learning Rate	1×10^{-3}
Final Learning Rate	1×10^{-5}
Batch Size	2048 sequences
Total Steps	1,000

new. Instead, our results provide a data-valuation view that helps explain why they can be useful in open-web pretraining. Large within-source utility variance suggests that the order and frequency with which different score regions are seen can matter; scorer bias further suggests that these schedules should be interpreted relative to the target capability profile of the recipe.

C Quantile Benchmarking Results and Discussions

C.1 Detailed Quantile Benchmarking Results

Quantile benchmarking settings. The from-scratch quantile probes use a Qwen3-0.6B-scale model and approximately 8.4B probe tokens per run. The resume probes start from a checkpoint trained on approximately 367B deduplicated DCLM tokens and then continue for approximately 8.4B probe tokens. It adopts the AdamW optimizer with weight decay, β_1, β_2 as in Table 17. The from-scratch run uses a constant LR schedule, following setup in Table 3, and the resume run uses a linear decay, resuming from a constant LR trained checkpoint, and decay to near-zero learning rate, shown in Table 3.

PreSelect scoring and deduplication. Nemotron-CC-v2 does not release aligned per-sample scores in the same way as DCLM or FineWeb-Edu. We therefore rescore it with PreSelect [64], using the lightweight released FastText model as a common proxy for its internal ordering. For DCLM and FineWeb-Edu, we first apply global fuzzy deduplication before quantile benchmarking. The quantiles are then computed on the deduplicated corpora, not on the original raw corpora. This is important because the score distribution and the retained token volume both change after deduplication. [5]

Quantile resolution and overhead. Because each probing chunk has a fixed token budget, the quantile width covered by one window depends on the total source size after deduplication. A

[5]The Nemotron-CC-v2 duplication rate is relatively smaller and we do not perform additional deduplication.

Table 4: From-scratch/resume ordering consistency over shared source-quantile units. We report Spearman rank correlation on the shared score quantiles 0, 30, 45, and 60.

Benchmark	Spearman	Interpretation
HSwag	0.91	highly stable
MMLU	0.85	highly stable
ARC-E	0.82	highly stable
SIQA	0.66	moderately stable
CSQA	0.36	weakly stable
ARC-C	0.04	unstable

smaller source is therefore probed at coarser quantile resolution than a larger source. In our main English experiments, each probing run uses approximately 8.4B tokens, and benchmarking five quantiles consumes 42B probe tokens in total for one dominant source. The main deduplicated candidate pools are still large: DCLM Baseline and FineWeb-Edu exceed 600B and 190B tokens in the relevant recipe tables, so one 8.4B-token probe covers only a small slice of each score distribution. We nevertheless treat this as a resolution limitation rather than a fully ablated design choice, and account for it when interpreting the discrete probes. The method is designed to be cheap enough for recipe development, especially for large web corpora, where a small change in retained fraction can move many tokens. 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 one-pass training on the dedup-DCLM Baseline dataset. In the final end-to-end application, the 0.6B-scale probing experiments for five DCLM quantiles account for less than 0.6% of the total pretraining budget in Section 7. We therefore view quantile benchmarking as a lightweight recipe-search overhead rather than a separate large-scale pretraining stage.

Ordering is reasonably stable across probing regimes. This stability is not uniform across benchmarks, and that is itself informative. Over the shared source-quantile units, from-scratch/resume Spearman rank correlation reaches 0.91 for HellaSwag, 0.85 for MMLU, and 0.82 for ARC-Easy. In contrast, it is only 0.36 for CSQA and 0.04 for ARC-Challenge. Some benchmarks are therefore much better anchors than others. This aligns with recent work on benchmark signal-to-noise ratio [29]. In particular, MMLU and ARC-Easy are not only knowledge-oriented, but also far more stable for recipe deduction. We summarize these values in Table 4.

Full quantile benchmarking plots. We provide the full quantile plots here. The main paper uses representative findings, while this appendix preserves the full visual evidence across benchmark families and probing regimes. Results in Figures 6 and 7.

C.2 Discussion: Connection to Prior Work

Quantile benchmarking and the controlled results also provide a common lens for understanding prior work.

Why a smaller filtering ratio does not contribute to better performance. Methods such as BETR [51] study how performance changes with different filtering ratios under a fixed scorer. They find that making the retained fraction smaller does not always improve average performance, especially when the retained fraction is already low. Quantile benchmarking gives one explanation. The highest-scored region can be biased toward some benchmarks, or it can be unstable. Filtering more aggressively can therefore improve one capability while hurting the average.

Why curriculum can work. Curriculum methods become more meaningful once the within-source utility variance is large. Recent work shows that curriculum learning can help when it is paired with a training dynamic that keeps late high-quality data from being under-trained [47]. Our findings explain why this condition appears in scored web corpora: even within strong curated sources, the gap between better and worse score quantiles can be several benchmark points. Moving high-utility partitions toward later training stages lets the model spend more final updates on data aligned with the target capability. This explains why the ascending curriculum still improves the controlled 40B

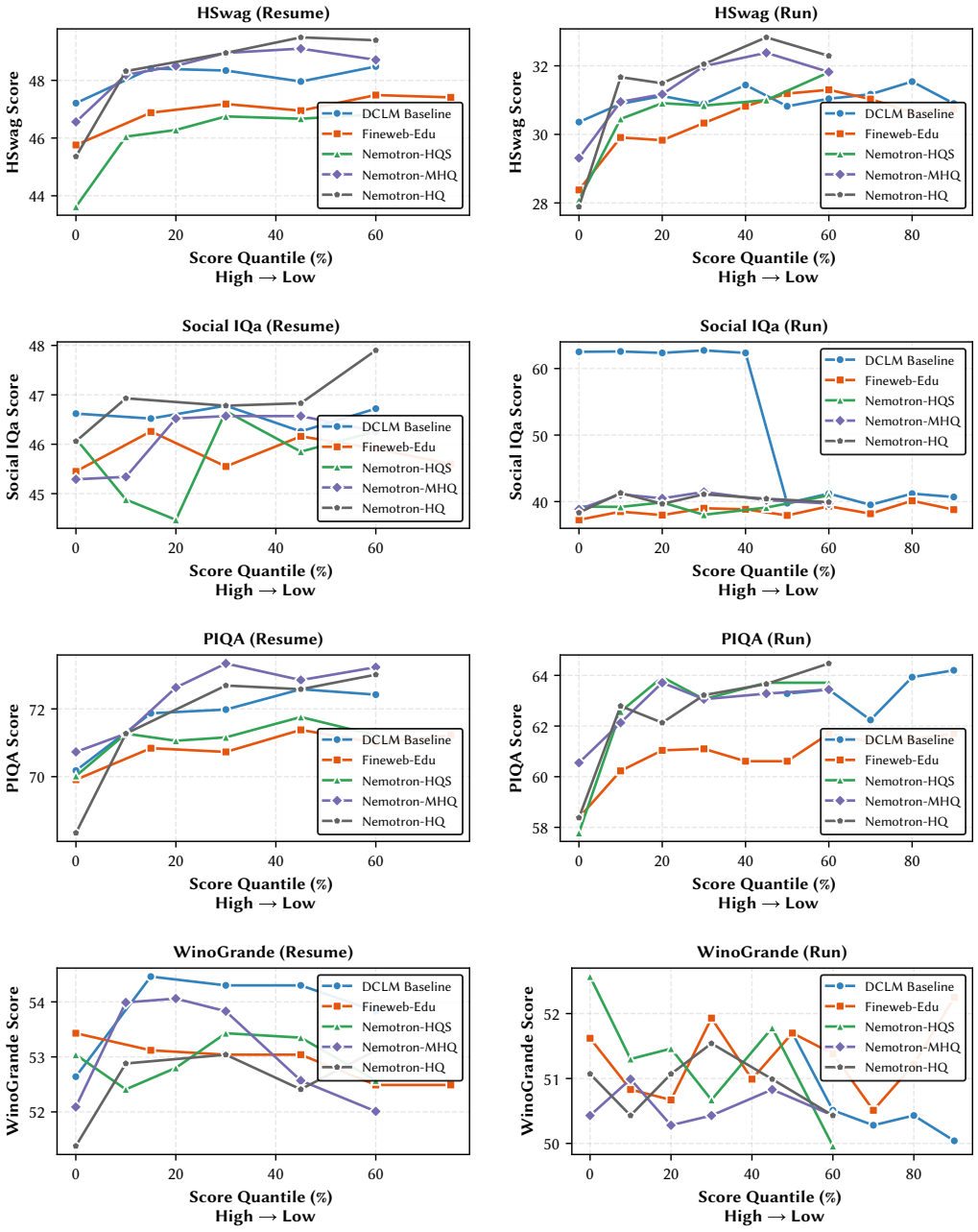

Figure 6: Understanding- and commonsense-oriented quantile plots.

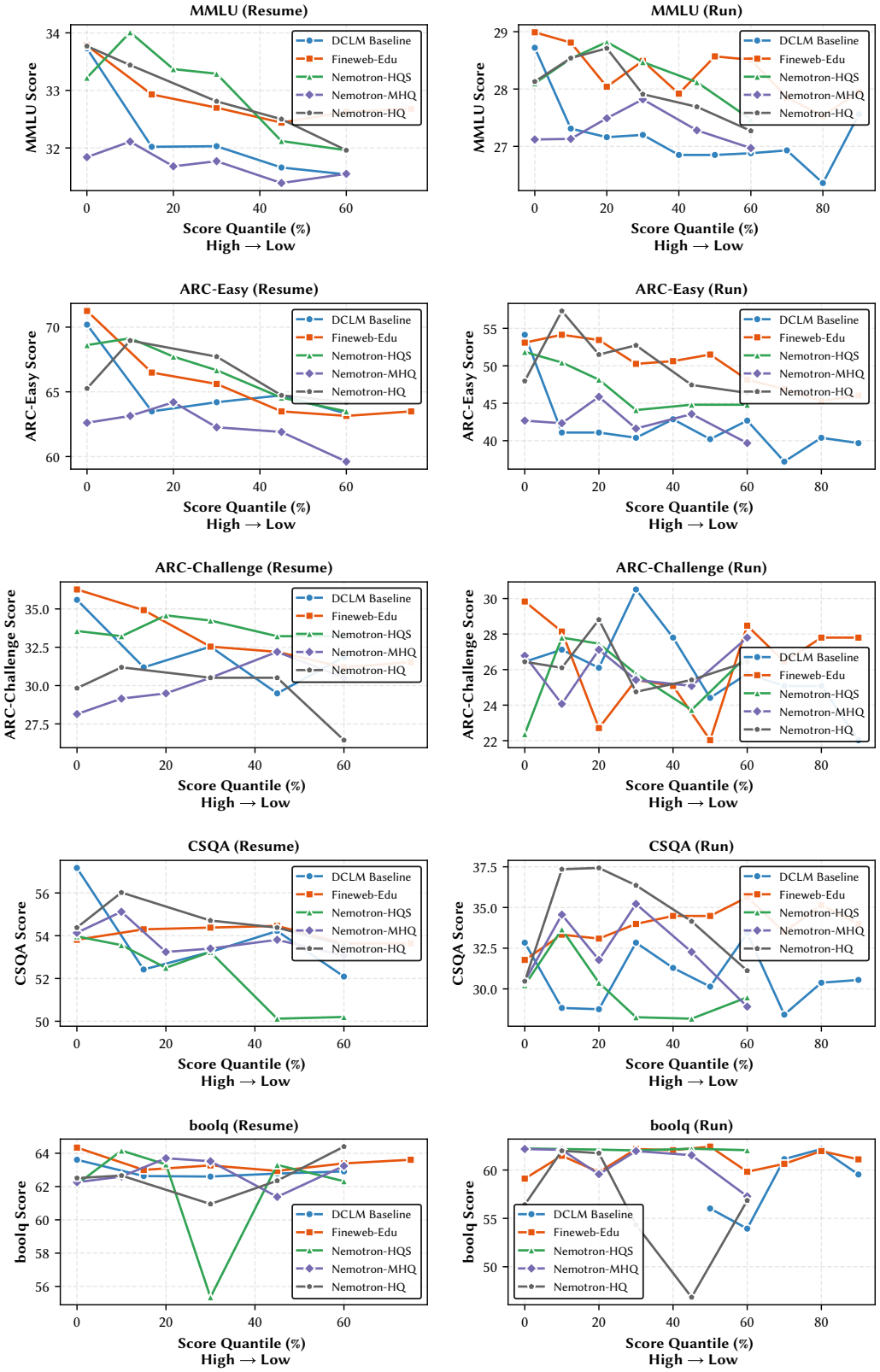

Figure 7: Knowledge-oriented quantile plots.

Table 5: Training hyperparameter configurations for the controlled 40B-token top-p mixing ablation and the 30B-token curriculum and selective repetition ablation.

Top-p mixing runs

Parameter	Value
Peak Learning Rate	1×10^{-3}
Final Learning Rate	1×10^{-5}
Global Batch Size	1024 sequences
Context Length	4096
Total Steps	10,000
Processed Tokens	41.94B
Decay Steps	2,500
Warmup Steps	500

Curriculum and selective repetition runs

Parameter	Value
Peak Learning Rate	3×10^{-3}
Final Learning Rate	1×10^{-3}
Global Batch Size	512 sequences
Context Length	4096
Total Steps	15,375
Decay Steps	2,875
Warmup Steps	768

ablation after the mixture has already been corrected. The curriculum pilot in Table 2 provides a small-scale reproduction of this effect.

Why repetition and D4-style designs can work. The same utility-distribution view also explains selective repetition and related designs such as D4 [69]. If the highest-utility portion of a source is concentrated in a relatively small tail, then repeating that tail can improve parameter efficiency under a fixed token budget. Repetition helps when the marginal utility of the best retained region is high enough to justify extra exposure. The selective-repetition pilot in Table 2 is consistent with this picture: mild repetition of a refined subset helps, while more aggressive repetition can over-specialize toward a narrow capability slice.

D Detailed Ablation Results

This section provides the full controlled ablation materials referred to in the main paper.

Experiments setup. The controlled ablations use the 2B-scale architecture family used in the final application run. The architecture follows Qwen3-1.7B but unties word embeddings, resulting in about 1.4B non-embedding parameters and about 0.6B embedding parameters. The LR schedule uses WSDSC [46]. All four top-p mixing runs use the same 10,000-step training schedule in Table 5. The main comparison changes only the data interface: uniform source sampling, vanilla top-p filtering, quantile-benchmarking-guided top-p filtering, and the same QB mixture with an ascending instance-level curriculum.

The ablation candidate pool is constructed before applying the four strategies. We uniformly sample approximately 50B tokens from each of FineWeb-Edu, DCLM Baseline, and Nemotron-CC-v2 (51.4B, 51.7B, 51.9B, respectively), and then attach the corresponding source-specific quality scores; Nemotron-CC-v2 uses PreSelect scores [64]. All percentages in Table 6 are token shares within the selected mixed pool before training-budget matching, while the top-p retained fractions are measured against these approximately 50B-token candidate pools. The vanilla top-p baseline keeps roughly equal token contributions from the three sources and fills each quota with the highest-scored tokens from the corresponding candidate pool.

In practice, we use top-p sampling on documents rather than directly on tokens, so there are some approximations. The detailed selection table sees Table 7, where we top-p sample from candidate documents and obtain the final mixtures.

Training-token accounting. After selection and preprocessing, the Baseline, Top-p Mixture, and QB/QB-Asc pools contain 44.10B, 42.56B, and 41.80B tokens, respectively. These are available pool sizes, not the amounts consumed by training. All four runs process $10{,}000 \times 1024 \times 4096 \approx 41.94 \times 10^9$ token positions: the Baseline and Top-p pools are downsampled, while the QB/QB-Asc pool is slightly upsampled to match this common budget. Training volume counts processed token positions, including repeated occurrences from upsampling, rather than unique pool tokens. We use "40B" as a rounded label for this shared training budget.

Table 6: Selected-pool compositions before training-budget matching. The Baseline, Top-p Mixture, and QB/QB-Asc pools total 44.10B, 42.56B, and 41.80B tokens, respectively. Token counts and shares describe these pre-resampling pools, not training consumption. Downsampling or upsampling matches all four runs to 41.94B processed tokens (nominally 40B; Table 5).

Configuration	Dataset	Sampling Method	Pool Tokens (B)	Pool Share
Baseline	FineWeb-Edu	Uniform	15.23	34.5%
	DCLM Baseline	Uniform	13.38	30.3%
	Nemotron-CC-v2	Uniform	15.49	35.1%
Top-p Mixture	FineWeb-Edu	Top-p Filtered	14.05	33.0%
	DCLM Baseline	Top-p Filtered	13.57	31.8%
	Nemotron-CC-v2	Top-p Filtered	14.94	35.1%
QB / QB-Asc	FineWeb-Edu	QB Threshold	16.14	38.6%
	DCLM Baseline	QB Threshold	4.05	9.7%
	Nemotron-CC-v2	QB Threshold	21.61	51.7%

Table 7: Data mixture construction for the controlled ablation experiments. Candidate tokens are sampled before thresholding, with approximately 50B tokens from each source. Doc top-p denotes the document-level retained fraction used to choose the threshold. Because the threshold is applied at the document level, the resulting token retention can differ from the nominal doc top-p. Mixture share is the token share within the selected mixed pool before budget matching; each run then downsamples or upsamples this pool to the same 41.94B processed-token budget (Table 6).

Configuration	Dataset	Candidate Docs (M)	Candidate Tokens (B)	Doc Top-p	Retained Tokens (B)	Token Retention	Mixture Share
Top-p Mixture	FineWeb-Edu	51.1	51.4	27%	14.1	27.3%	33.0%
	DCLM Baseline	40.8	51.7	33%	13.6	26.2%	31.9%
	Nemotron-CC-v2	66.2	51.9	31%	14.9	28.8%	35.1%
QB / QB-Asc	FineWeb-Edu	51.1	51.4	31%	16.1	31.4%	38.6%
	DCLM Baseline	40.8	51.7	13%	4.1	7.8%	9.7%
	Nemotron-CC-v2	66.2	51.9	41%	21.6	41.6%	51.7%

The curriculum and selective repetition experiments use the Qwen2.5-1.5B model without shared embedding and are trained on 30B tokens. The training hyperparameter setup detail is also shown in Table 5. For the curriculum averaging setup, we use a 0.21B checkpoint interval and average over the last 6 checkpoints.

For evaluation, performance gains are particularly notable in the "Core" set[6], which consists of high-signal benchmarks used to distinguish model capabilities [29].

E Curriculum and Selective Repetition Details

This section records how we implement curriculum and selective repetition in the fully open KAIYUAN-2B application run and how they connect to the quantile-benchmarking view.

E.1 Multi-Dataset Curriculum Construction

The goal of the instance-level curriculum is simple: keep the phase-level source mixture stable, but present higher-scored samples later inside each phase. This matters because late training steps and late checkpoints have a strong effect on the final model. If source mixtures were implemented by concatenating datasets, the model would see large distribution jumps. We instead interleave sources after normalizing their within-source ranks.

For each source D_i, let $r_i(x)$ denote the local rank of sample x after sorting by the source's own quality score in ascending order. We compute a global rank

$$R(x) = r_i(x) \cdot \frac{N_{\text{total}}}{|D_i|},$$

[6]including MMLU [31], ARC-Easy and ARC-Challenge [17], CSQA [68]

Table 8: Detailed downstream evaluation results for the controlled ablation configurations. Avg. averages all eight visible benchmark columns in this table. Core averages MMLU, ARC-Challenge, ARC-Easy, and CSQA. Avg. and Core subscripts report the absolute margin over the uniform baseline. Positive margins are shown in green.

Method	Mixture	Curriculum	MMLU	ARC-c	ARC-e	BoolQ	CSQA	PIQA	SIQA	Wino.	Avg.	Core
Baseline	Uniform	Random	34.49	34.58	72.84	64.95	59.79	74.27	**48.67**	54.14	$55.47_{0.00}$	$50.43_{0.00}$
Top-p Mixture	Vanilla Top-p	Random	35.79	36.95	71.60	64.34	**60.28**	73.39	47.70	**54.30**	$55.54_{+0.07}$	$51.16_{+0.73}$
QB	Quantile-Based	Random	36.51	36.95	74.60	63.39	59.62	73.07	48.16	53.12	$55.68_{+0.21}$	$51.92_{+1.49}$
QB-Asc	Quantile-Based	Ascending	**36.65**	**38.64**	**75.66**	**65.57**	58.72	**74.92**	47.44	53.83	$\mathbf{56.43}_{+0.96}$	$\mathbf{52.42}_{+1.99}$

Table 9: Selective repetition and curriculum-learning pilot results.

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

where $N_{\text{total}} = \sum_i |D_i|$. In this equation, D_i denotes the retained partition of source i used in the current phase, not the original unfiltered corpus. Sorting all samples by $R(x)$ gives a globally interleaved curriculum. The key property is that a phase can maintain its target mixture ratios while still making the average local quality increase over training time. This is the implementation behind the "ascending" curriculum used in the controlled ablation of Tables 1 and 8.

E.2 Phase-Wise Selective Repetition

Selective repetition uses the same utility-tail intuition as threshold deduction. If the highest-scored part of a source has much higher marginal utility than the rest, then a fixed token budget should not spend equal exposure on all regions. Instead, later phases can retain a smaller and higher-scored subset, so the best tail receives multiple exposures while the lower-scored tail is seen fewer times.

The FineWeb-Edu schedule in Figure 9 illustrates the exposure pattern. The top 10% region appears in Phases 2–5. The 10–30% band appears in Phases 2–4. The 30–50% band appears in Phases 2–3, and the remaining 50–100% region appears only in Phase 2. This practice follows prior work like SmolLM2 [3]: later exposures operate on a smaller and higher-utility tail.

The pilot results in Table 2 show why the schedule should be moderate. Repeating a refined subset can improve parameter efficiency, but overly aggressive repetition can over-specialize toward a narrow capability slice. In the final recipe, we therefore cap repetition by the number of training phases and use repetition as a controlled extension of the threshold view rather than as an unconstrained data amplification trick.

E.3 Instruction-Style Data in Late Pretraining

The final KAIYUAN-2B recipe includes a small amount of instruction-style and downstream-style data in the last phases, as recorded in Tables 15, 16 and 18. This is a common practical choice in recent open recipes: late instruction-style mixtures can act as a warmup for later supervised tuning, reduce the distribution shift between base pretraining and downstream adaptation, and stabilize behavior on structured prompt formats. YuLan and OLMo-style open recipes similarly include curated high-value or instruction-like components in late training stages [34, 70, 95]. For fairness, we have excluded all evaluation test splits from the downstream-style training mixture.

This design also creates an attribution caveat. The 2B release case should not be read as a pure causal ablation of open-web quantile benchmarking alone. It is an end-to-end application recipe that combines threshold-guided web selection with phase-wise domain mixtures, selective repetition, curriculum, and small late-stage instruction-style components. The controlled 40B-token ablation in Tables 1 and 8 is the cleaner evidence for the effect of quantile-guided top-p selection. For fairness, the comparison tables use base checkpoints without separate post-training.

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	–	–	34.63	18.90	–
Llama-3.2-3B	3B	45.67	44.33	–	–	49.42	29.88	–
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	–	–	15.56	6.71	–
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 base-model reports or original papers; their evaluation settings may differ from the shot counts shown, which apply to our evaluations.
– denotes an unverified base-model result. Averages are reported only for models with results on all six tasks; instruction-tuned results are not substituted for missing base-model results.

Table 11: Performance across reasoning and knowledge benchmarks.

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

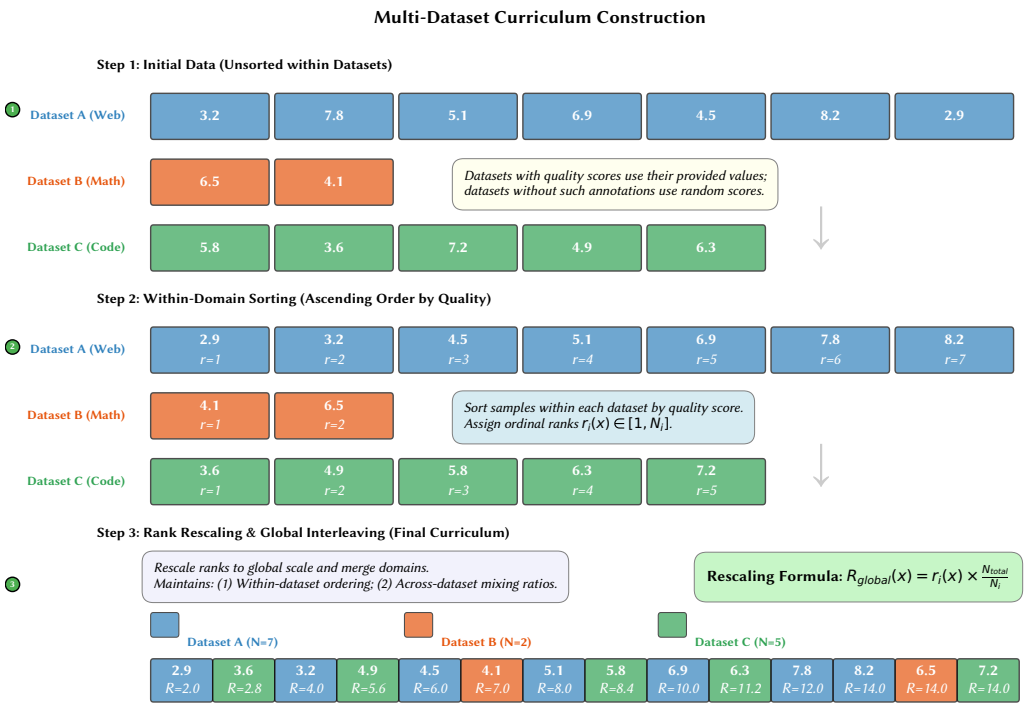

Figure 8: **Multi-dataset curriculum construction.** This schematic shows how we build an instance-level curriculum when sources have different local quality scores. We first sort each source by its own score. We then rescale each sample's local rank to a shared global rank and interleave the sources by that rank. The procedure preserves the intended source mixture while moving higher-scored samples toward later training steps. Sources without quality scores are shuffled by assigning random ranks.

F Utility Fitting and Mixture Deduction

Monotone Fitting Details The threshold-deduction step needs an inverse map from benchmark utility to a retained top-p fraction. Raw quantile probes do not always provide such a map directly because the observed benchmark values can be noisy. We therefore fit a conservative monotone surrogate before interpolation.

For one source and one anchor benchmark, let the probed points be $\{(q_j, y_j)\}_{j=1}^{K}$, sorted by increasing retained fraction q_j. Larger q_j means that the retained subset reaches deeper into lower-scored data. We solve

$$\min_{\hat{y}_1,\ldots,\hat{y}_K} \sum_{j=1}^{K} (\hat{y}_j - y_j)^2$$

subject to

$$\hat{y}_j - \hat{y}_{j+1} \geq \epsilon, \quad j = 1, \ldots, K-1.$$

We use $\epsilon = 10^{-3}$ in the plotting and threshold scripts. This small margin removes flat inversions without materially changing benchmark values. After obtaining the fitted values $\hat{y}_j$, we build a shape-preserving PCHIP interpolator over $(q_j, \hat{y}_j)$. For thresholds outside the probed range, we use linear head or tail extrapolation with a non-positive slope bounded by $-\epsilon$. The inverse is then clamped to the valid interval $[0, 100]$: thresholds above the best attainable fitted utility map to 0%, and thresholds below the worst full-source utility map to 100%.

The threshold T^* is then found by grid search. The retained fraction for each source is obtained by inverting its fitted curve at T^*. In the controlled 40B-token ablation, the token volumes N_i are the approximately 50B-token candidate pools sampled for that ablation, not the full raw corpus sizes. This convention matches the ablation table and avoids a misleading comparison in which uniform and heuristic top-p baselines would sample only from the absolute tip of multi-trillion-token corpora.

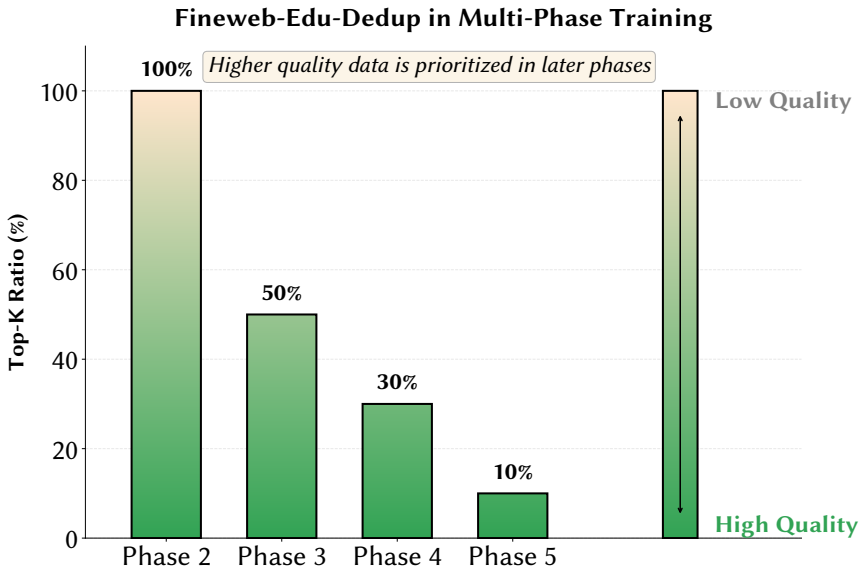

Figure 9: **Selective repetition through phase-wise top-p schedules.** The plot shows the FineWeb-Edu-Dedup schedule used as a representative example. Phase 2 uses the full retained source. Later phases progressively narrow the retained subset to the top 50%, 30%, and 10%. As a result, the highest-scored 10% tail is reused across all later phases, while lower-scored regions receive fewer exposures.

Discussion on top-p mixing assumptions. The formulation of top-p mixing relies on two practical assumptions. First, the local scores must contain some useful signal: retaining a higher-scored top-p fraction should help at least one capability dimension. Second, the sources must share at least one target capability dimension that can serve as a common utility metric. If lower-scored regions are consistently better, or if the score distribution has no relation to downstream utility, then threshold deduction is not the right tool. In that case, quantile benchmarking is still useful as a diagnostic, but it should not be used as an optimization procedure without redesigning the scorer or target metric.

Boundary cases in threshold. For completeness, we clamp the inverse map at the feasible boundaries. If the threshold is higher than a source can reach, the mathematical solution gives $q_i^* = 0$, meaning that the source contributes no tokens under the strict threshold rule. If the threshold is lower than the whole source tail, we set $q_i^* = 100$. In practical release recipes, one may still keep a small predefined floor, such as the top 2–5%, for diversity or coverage reasons. We do not use such a floor in the controlled threshold ablation, so the mixture reported there follows the strict clamped rule.

Anchor robustness. The anchor benchmark matters, but the resulting mixture is not arbitrarily fragile. In additional robustness tests, we move the anchor gradually from pure MMLU to pure ARC-Easy. We use four anchors: MMLU, $0.7 \cdot \text{MMLU} + 0.3 \cdot \text{ARC-Easy}$, $0.5 \cdot \text{MMLU} + 0.5 \cdot \text{ARC-Easy}$, and ARC-Easy. The robustness computation uses the same fitted threshold-deduction pipeline as Figure 3: DCLM and FineWeb-Edu use probes at top-$\{0, 15, 30, 45, 60\}\%$, while Nemotron-HQ uses top-$\{0, 10, 30, 45, 60\}\%$ because its available probe grid contains top-10% rather than top-15%. Figure 10 summarizes this behavior. The retained fractions change in a systematic way, but the ordering remains stable: DCLM is truncated most aggressively, FineWeb-Edu is intermediate, and Nemotron-HQ is retained deepest. The robustness test therefore supports a structural claim: the method captures a stable ordering of source tails, not a brittle threshold chosen from one benchmark.

G Recipe Details and Phase-wise Mixtures

This section records the concrete phase-wise recipe used in the KAIYUAN-2B application run. The main paper treats this as a practical realization of the threshold view rather than as a primary scientific contribution.

Table 12: 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 13: 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 14: 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%

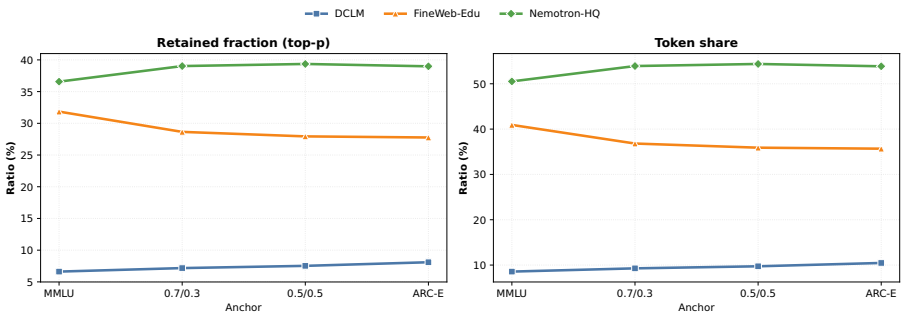

Figure 10: **Anchor robustness of threshold deduction.** We recompute threshold-based mixtures under four anchors, ordered from pure MMLU to pure ARC-Easy, using the same monotone fitted threshold rule as in Figure 3. Retained fraction is the top-p percentage kept within each original source dataset, so its denominator is the source's own token volume. Token share is the percentage of the final 40B-token mixture contributed by that source. The exact proportions change across anchors, but the source ordering remains stable: DCLM is shallowest, FineWeb-Edu is intermediate, and Nemotron-HQ is deepest.

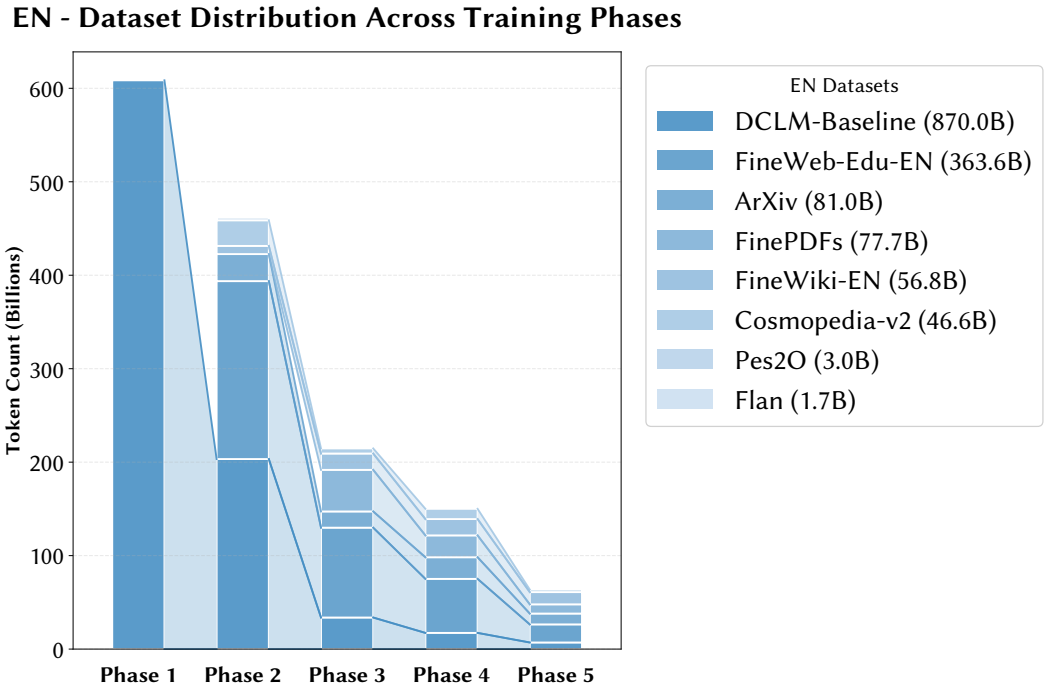

Figure 11: Phase-wise mixture: English sources.

H Training and Evaluation Details

H.1 Training Configuration

In Table 17, we present the details of our training hyperparameter configuration in three parts:

- For the model architecture, we primarily follow Qwen3-1.7B [83] and adopt the vocabulary from the Qwen series [81, 83]. We use $\theta = 10000$ for RoPE [67] to support a context length of 4K. The Soft-Capping threshold is set to 30.0.
- 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.

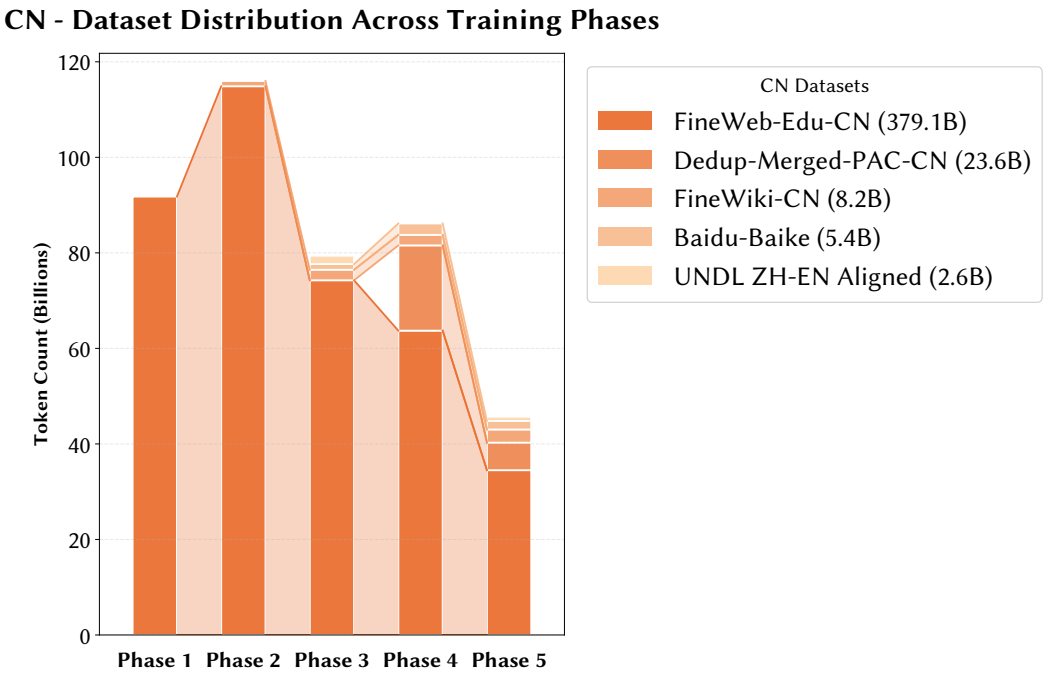

Figure 12: Phase-wise mixture: Chinese sources.

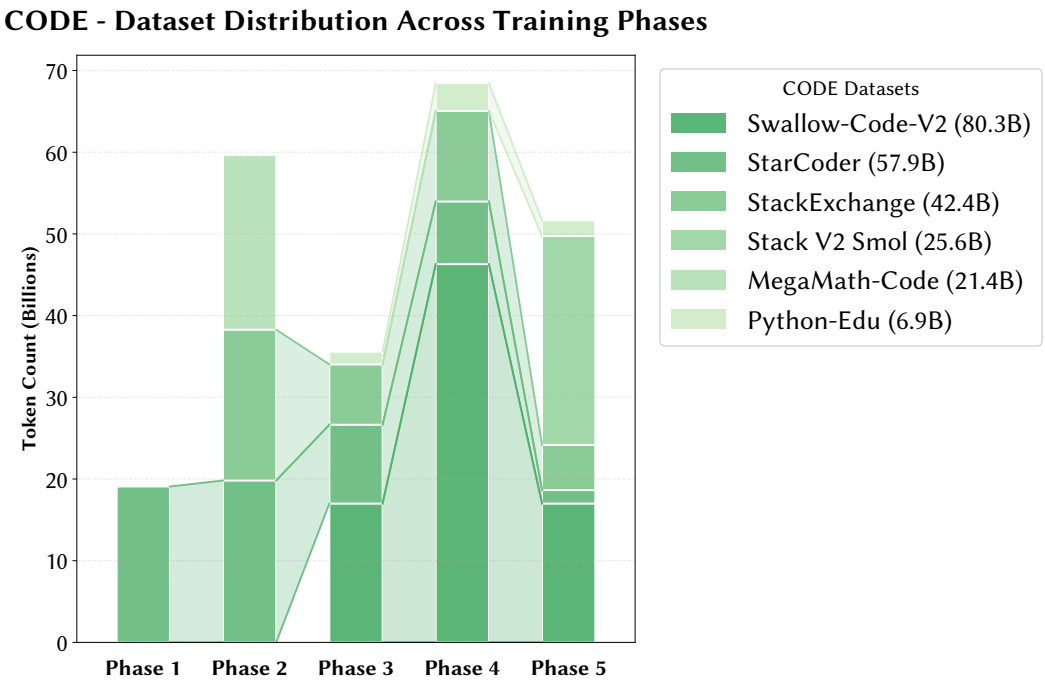

Figure 13: Phase-wise mixture: code sources.

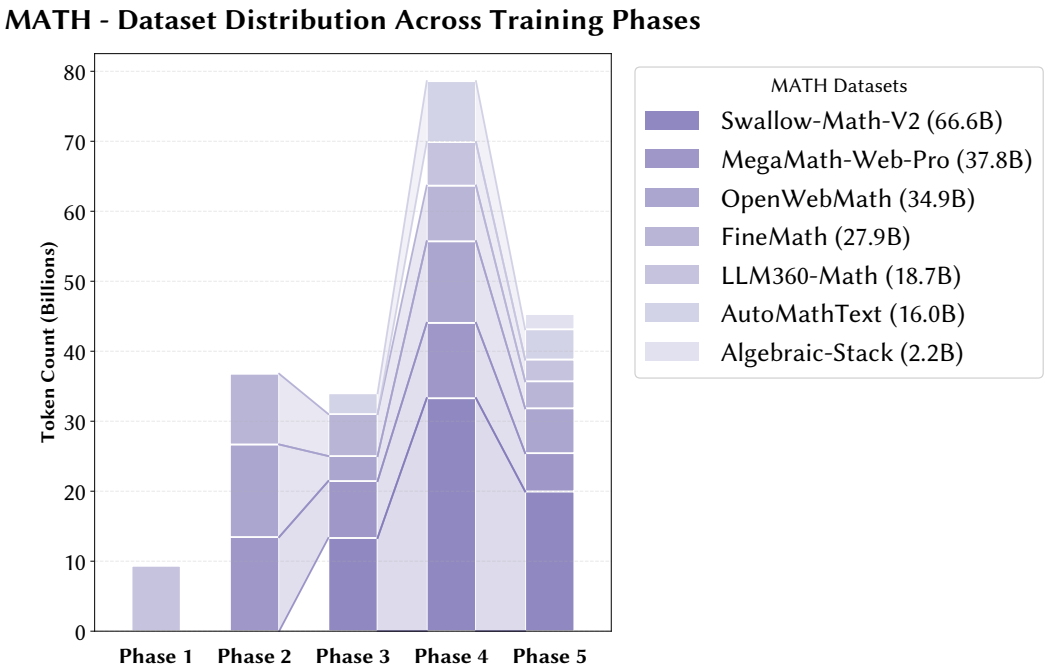

Figure 14: Phase-wise mixture: math sources.

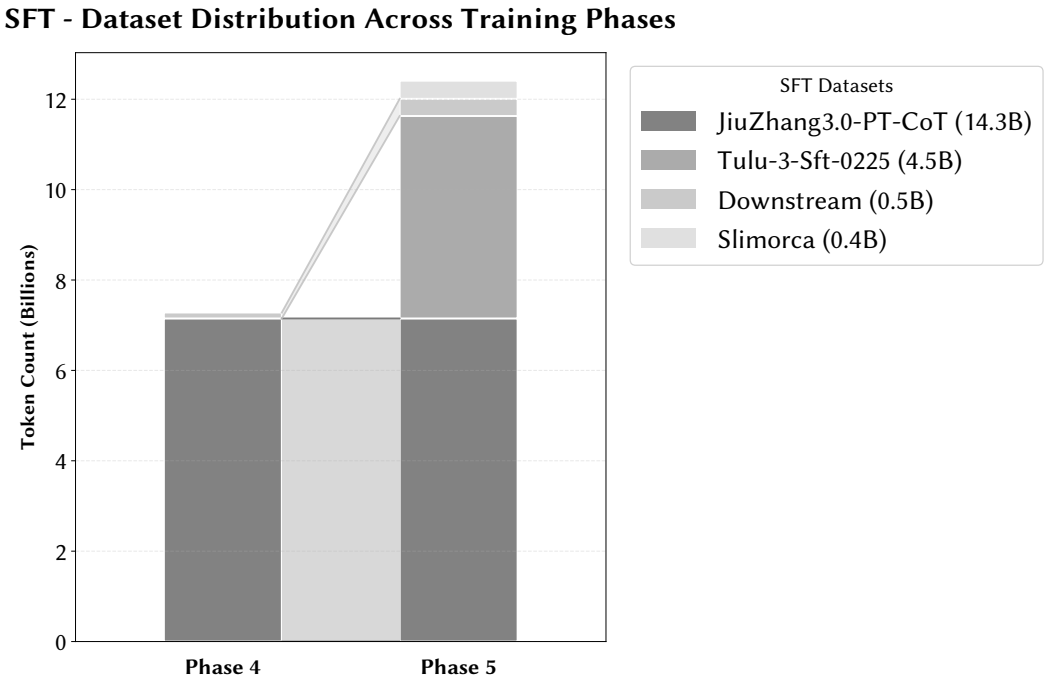

Figure 15: Phase-wise mixture: supervised signals.

Table 15: 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%

- 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.

H.2 Evaluation Setup

H.2.1 Base Model Evaluation Setup

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.[7]

Open-weight models. This group includes Qwen2-1.5B [81] (7T tokens), optimized for multilingual and coding tasks; the Qwen2.5 series [82] (1.5B and 3B, 18T tokens), featuring refined architectures for mathematical reasoning; and the Qwen3 series [83] (0.6B to 4B, 36T tokens), supporting extended contexts. We also include Gemma2-2B [60], distilled using 2T tokens, and the Llama3.2 series [49] (1B and 3B, 9T tokens), designed for on-device inference.

Fully open models. We compare against SmolLM2-1.7B [3] (11T tokens), which utilizes the Llama 2 architecture; SmolLM3-3B [8], a data-centric model trained on 11T tokens; OLMo2-1B [70] (4T tokens); and YuLan-Mini [85], a 2.4B-parameter model optimized for data efficiency using 1.1T tokens, with handcrafted fine-grained 28-phase training.

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

Table 16: 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 17: 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

H.2.2 Benchmarks

Evaluation spans four primary domains. **Mathematics** is assessed via GSM8K [18] and MATH [32]. **Coding** proficiency is measured using the MBPP [6] sanitized subset and HumanEval [15]. **Chinese language processing** utilizes CMMLU [40] and C-Eval [35]. Finally, **General Reasoning & Knowledge** is assessed via a suite of eight benchmarks: MMLU [31], HellaSwag [88], CSQA [68], BoolQ [16], PIQA [14], SocialIQA [62], WinoGrande [61], and ARC [17].

H.2.3 Implementation Details

Evaluations are conducted using the OpenCompass framework [22]. Mathematics and coding tasks use *generation mode*[8], while other benchmarks employ *perplexity-based (PPL) evaluation*. Following the OLMES protocol [25], PPL tasks are assessed under both multiple-choice (MCF) and completion formulations (CF), with the higher score reported.

Official generation results are included only when they can be attributed to the base checkpoint; their original evaluation settings may differ from ours. We omit unverified base-model GSM8K and MATH results for Llama-3.2-1B, Llama-3.2-3B, and OLMo-2-0425-1B rather than substituting instruction-tuned results. These missing entries are marked – in the tables. Six-task and fifteen-task averages are reported only for complete rows, and the parameter-efficiency plots include only models with a complete fifteen-task average. The nine-task reasoning and knowledge comparison is unaffected.

I Release and Dataset Details

Public release. The KAIYUAN-2B release is hosted under the full name `PCMind-2.1-Kaiyuan-2B`, consistent with the public technical report. Model weights and the model card are available at `https://huggingface.co/thu-pacman/PCMind-2.1-Kaiyuan-2B`. The phase-wise training dataset and its reconstruction instructions are available at `https://huggingface.co/datasets/thu-pacman/PCMind-2.1-Kaiyuan-2B`. The preprocessing framework is available at `https://github.com/thu-pacman/Kaiyuan-Spark`, and the training framework is available at `https://github.com/thu-pacman/kaiyuan-mindformers` on the `kaiyuan` branch.

The fully open release case prioritizes redistributable datasets and reproducible preprocessing. This means that not every strong source discovered during quantile benchmarking can be used in the final released recipe. In particular, Nemotron data are excluded from the final open recipe because their license does not permit unrestricted redistribution.

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 [41]	4T	CC BY 4.0[1]
FineWiki-EN	English	HuggingFaceFW/finewiki [57]	8.7B	CC BY-SA 4.0[6]
FinePDFs	English	HuggingFaceFW/finepdfs [38]	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 [12]	220B	ODC-By 1.0[1]
ArXiv	English	togethercomputer/RedPajama-Data-1T [21]	28B	Metadata: CC0 1.0 [5] Content: various [4]

Continued on next page

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

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

Name	Type	Hugging Face ID	#Tokens[0]	License(s)
Cosmopedia-v2	English	HuggingFaceTB/smollm-corpus [12]	27B	ODC-By 1.0
FineWiki-CN	Chinese	HuggingFaceFW/finewiki [57]	1.1B	CC BY-SA 4.0[6]
Fineweb-Edu-CN	Chinese	opencsg/Fineweb-Edu-Chinese-V2.1 [86]	1.5T	OpenCSG Community License [20], 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 [72] Skywork/SkyPile-150B [76] OpenDataLab/WanJuan1.0 [27, 28][5] BAAI/IndustryCorpus BAAI/IndustryCorpus2 [63] WuDaoCorpus2.0 [91, 92][5]	178B	CCI{,2,3}-Data: CCI Usage Agreement [55] SkyPile-150B: Skywork Community License [65], 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 [56]	14.7B	ODC-By 1.0[1]
FineMath	Math	HuggingFaceTB/finemath [3]	10B	ODC-By 1.0
MegaMath-Web-Pro	Math	LLM360/MegaMath [93]	300B	ODC-By 1.0
AutoMathText	Math	math-ai/AutoMathText [90]	8.7B	CC BY-SA 4.0
SwallowMath-v2	Math	tokyotech-llm/swallow-math-v2 [23]	32B	Apache 2.0
StarCoder	Code	bigcode/starcoderdata [37]	250B	Original Licenses[2]
Stack V2 Smol	Code	bigcode/the-stack-v2 [45]	900B	Original Licenses[2]
StackExchange	Code	togethercomputer/RedPajama-Data-1T [21]	20B	CC BY-SA 2.5/3.0/4.0[3]
Python-Edu	Code	HuggingFaceTB/smollm-corpus [12, 45]	3.4B	ODC-By 1.0, Original Licenses[2]
Algebraic-Stack	Code	typeof/algebraic-stack [7, 56]	11B	ODC-By 1.0[1]
Swallow-Code-v2	Code	tokyotech-llm/swallow-code-v2 [23]	49.8B	Apache 2.0
SlimOrca	SFT	Open-Orca/SlimOrca [44, 53]	190M	MIT
JiuZhang3.0-Corpus-CoT	SFT	ToheartZhang/JiuZhang3.0-Corpus-CoT [94]	358B	*Not Specified*
Tulu-3-Sft-0225	SFT	allenai/tulu-3-sft-mixture [39]	640M	ODC-By 1.0 (mixed)
downstream[4]	SFT	cais/mmlu [30, 31] openai/gsm8k [18] allenai/ai2_arc [17] allenai/openbookqa [50] Rowan/hellaswag [88] allenai/winogrande [61]	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 [19].
[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 contents of Wiki are limited to license under CC BY-4.0 and GFDL.

J Full Benchmark and Parameter Tables

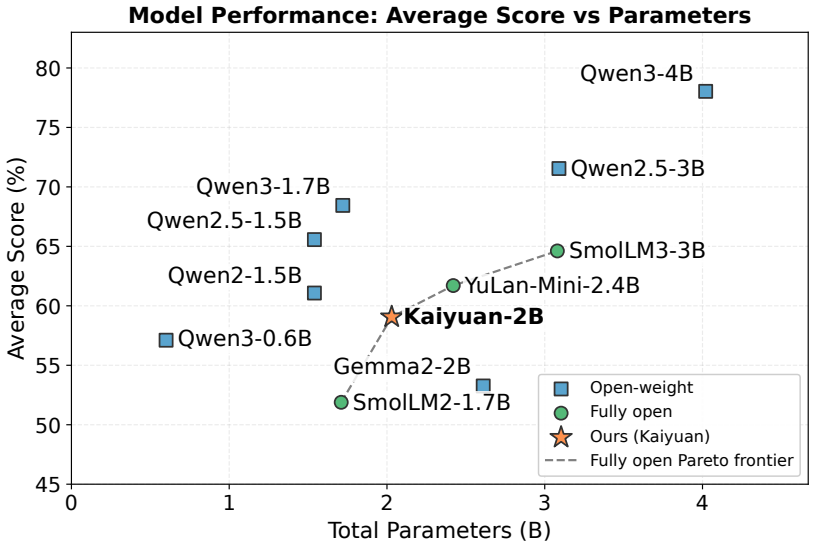

Figure 16: **Performance over total parameters.** KAIYUAN-2B remains on the frontier of compared fully open models under a total-parameter view. Only models with a complete fifteen-task average in Table 20 are plotted.

Table 19: 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 20: Full benchmark comparison across all reported evaluation tasks. – denotes an unverified base-model result. Avg. averages all fifteen tasks and is omitted for incomplete rows; instruction-tuned results are not substituted for missing base-model results. Result provenance and evaluation settings follow Table 10 and § H.2.

Model Name	Params	Math		Code		Chinese		Reasoning & Knowledge									Avg.
		GSM8K	MATH	MBPP (sanitized)	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	–	–	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	–
llama-3.2-3B	3B	–	–	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	–
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	–	–	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	–
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

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our contributions lie in problem formulation of heterogeneous top-p mixing, quantile benchmarking framework, some interesting and useful findings, validation ablations, and a fully open pretraining recipe for the community. See the abstract and Section 1

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Due to the pretraining cost, we can not ablate the benefit of each component from the end-to-end final run, and we can not afford a multi-seed run in validation experiments to report the error bound. Other limitations include benchmark dependence and release-compatibility constraints. Details in Section A.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [No]

Justification: The paper contains formal problem formulations and deduction equations in Sections 3, 6, and Appendix F, but these are operational formulations rather than theorem-proof style theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide training hyperparameters, model architecture, sampling, evaluation, and phase-wise mixture details in Sections C.1, D, G and H. Public model weights, data, preprocessing code, and training-code links are provided in Section I.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Model weights, phase-wise training data, and preprocessing and training code are publicly available through the links in Section I. We do not provide a complete executable reproduction package covering every quantile probe and controlled ablation, so we retain a No answer.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide setup details, from training hyperparameters (Sections C.1, D and H, Tables 3 and 5), model architecture (Sections C.1 and D), data sampling details (Table 7), evaluation setup (Section H.2), and data mixture document (Section I and tables 12 to 16).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: The main results are reported as single-run pretraining outcomes and controlled ablations without error bars. Section A explicitly notes the limited multi-seed validation at large scale.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The paper reports model size and token budget in Sections C.1, D and H), but it does not yet provide a complete per-experiment resource accounting with runtime and full memory details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have checked.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section A discusses positive impacts for reproducible open research and negative impacts such as misuse, inherited bias, and benchmark-driven recipe bias.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: The paper discusses release constraints and licensing-based exclusions, especially for Nemotron data in Appendix I.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Section I lists the datasets used in the release case, their sources, and their licenses, and the paper explicitly excludes non-redistributable sources from the final open recipe.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The released model and dataset include public model and dataset cards, usage and reconstruction instructions, and license information; the code repositories include their accompanying documentation (Section I). Training details and limitations are discussed in Sections A and H.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: No crowdsourcing or human-subject experiments.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No crowdsourcing or human-subject research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as a non-standard component of the core scientific method; AI assistance was limited to writing/editing support and is disclosed in Section A.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.